



Semantic Web

Towards a Semantic Web based Approach to Autonomic Management

Arun Kumar¹, Himanshu Chauhan¹, D. Janakiram²

¹IBM Research -India, 4, Block -C, Institutional Area, Vasant Kunj, New Delhi -110 070, India, USA

²Dept. Of Comp.Sc. & Engg, Indian Institute of Technology Madras, Chennai-600036, India

Abstract — Rapid growth in IT systems of various organizations built on heterogeneous technologies bundled together is leading to a management nightmare. It is becoming increasingly difficult to maintain these heavyweight IT systems that run the core functions of various organizations today. In this paper we propose to adopt an autonomic management approach to make complex IT systems self manageable. We propose a Semantic Web based solution approach that models managed entities as semantic web services.

I. INTRODUCTION

IT systems of various organizations typically grow in a haphazard manner while responding to business requirements. This eventually leads to a situation where heterogeneous technologies are patched together resulting in a management nightmare. It is becoming increasingly difficult to maintain these heavyweight IT systems that run the core functions of various organizations today. There is a need for new technologies that can deal with heterogeneity, unpredictability and scale of systems that are being deployed today.

There are different aspects of this problem that need effective solutions. These include the following:

The Managed Entities: This deals with issues related to managed entities themselves. The managed environment might include hardware elements such as routers, printers etc. or software elements such as web servers, databases, or both. It could also have metaelements such as the management applications themselves. Most efforts in the past such as Simple Network Management Protocol (SNMP) and Java Management Extensions (JMX) focused on one kind of element or the other. Recently, standards such as Common Information Model (CIM) have attempted to unify management of all elements of an IT environment.

Representation of Managed Entities and Management Information: Different representation technologies have been used in the past for describing managed entities. These include OSI's Abstract Syntax Notation One (ASN.1) used in SNMP, UML based graphical notation and Managed Object Format (MOF) used in CIM, and Managed Bean (MBean) in JMX etc. The design of these representations is driven by and hence,

limits the managed objects that can be efficiently represented in the system. A related issue is to have a representation of management information other than the entities themselves. This could include management policies, business rules, domain knowledge, etc. Rich representation of such information enables effective management techniques. However, richness needs to be traded with other design factors such as performance and ease of use.

The Management System: The management system could vary from an administrator monitoring alarms to an intelligent system that observes patterns in the collected data, correlates it across different components and detects failures, intrusions etc. The system needs to be scalable to perform reliably not only in normal situations but also, and more importantly, in very adverse situations leading to all sorts of alerts and cascading failures. With recent advances in global infrastructure utilization, the management system may have to deal with managed entities sitting in different administrative domains, across geographies and in heterogeneous environments.

Techniques for Effective Management: Given the scale of IT operations in various organizations today, it is highly desirable that as much as possible, the management tasks should be automated. This includes not only identifying abnormal situations but also decision making capability to deal with them autonomously. Such techniques that allow the system to be self-configurable, self-healing etc. are the focus of autonomic computing initiatives that have gained momentum in the recent past.

In this paper, we propose to adopt an autonomic management approach to make complex IT systems self manageable. We propose a Semantic Web based solution approach that models managed entities as semantic web services.

II. SERVICE ORIENTED APPROACH TO SYSTEMS MANAGEMENT

Various programming models have been proposed in the past for building distributed systems. Among these some of the popular ones included CORBA, DCOM, and Java RMI etc. Recently, Web Services [1] have gained acceptance as the technology of choice for building distributed systems. They separate end

point interface from their implementation and also, enable loose coupling between participating entities. A standardized message format (SOAP [0]) and a standard description language for specifying end point interface (WSDL [0]) establishes the basic protocol for interaction. XML is used as a vehicle for transporting messages in a platform independent manner. Further, using ontologies and AI concepts semantics have been introduced into web service descriptions resulting in languages such as OWLS [5]. This enables automatic web service discovery, invocation, composition and monitoring through programs that can reason.

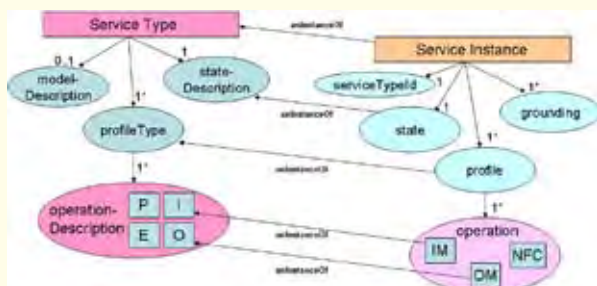


Fig. 1. Service Types and Service Instances

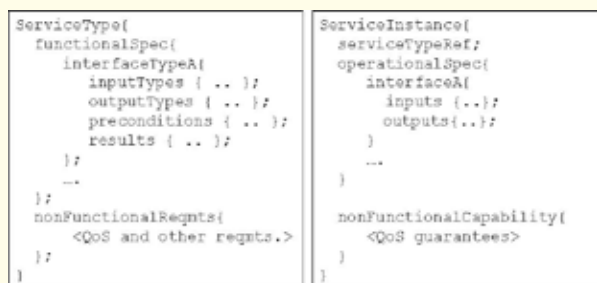


Fig. 2. Service Types and Service Instances

In [10], we presented an OO based semantic model for SOC as a unifying mechanism to represent all artifacts as services in an IT environment. It provides a representation for services that is rich in semantics and incorporates the abstraction principles of Classification, aggregation (service composition), interface inheritance and polymorphism in the context of SOC. Figure 1 and Figure 2 show the pictorial and pseudocode representation of Service Types and Service Instances respectively.

Service Type is a semantic specification of the service consisting of one or more profileTypes (i.e interface descriptions), an optional description of the process model and a description of *internal* state maintained by the service. Service Instance, on the other hand, is an operational specification of the service consisting of a reference to the Service Type, one or more profiles (i.e. interfaces), its internal state and a grounding.

III.A NEW PROGRAMMING MODEL FOR SERVICES

Having proposed a unifying mechanism to represent all artifacts as services in an IT environment, the next step is to facilitate discovery and use of artifacts that are described using this services based representation.

Towards that end we inspect the current services based software development methodology followed. We find that two different models of development are emerging. In one, developers need to program software agents that accept the requirements from the end user. The services needed to fulfill those requirements are then automatically discovered, selected, composed and invoked by these agents. The work being done by Semantic Web Services community plays an important role towards enabling this vision. While this model matures, the other model being employed by application developers is along the lines of traditional software development. Developers code enterprise systems by first developing new web services or by building wrappers for legacy systems or by using existing known services as components in their programs.



(a)

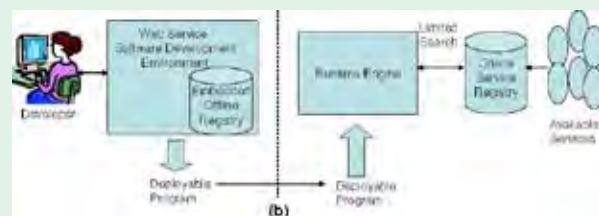


Fig. 3. Services based Software Development Model (a) Current (b) Proposed

Figure 3 (a) depicts the current Web Services based software development model in which the developer makes use of web services that are currently available for invocation. However, some of these services might not be available for use when this program is actually executed making the program brittle. When software agents are used, the desired web service can be searched at runtime but it adds a significant delay into the service invocation path. There is a tradeoff between program correctness and performance here. In both cases, the service invocation may not succeed even though a suitable service might be available to serve the request.

Current approaches are restrictive and facilitate strict matching of capabilities with requirements [6]. Even though approaches such as [7] use information retrieval techniques, and [8, 9] allow a softer notion of matching, each of these are limited in their effectiveness. We illustrate this with an example. Consider the service descriptions shown in Figure 4, described in terms of a simplified representation of their Input, Output, Preconditions and Effects. Here, two services -FreshFlowerShop service and FragrantFlowerShop service are offering similar functionality, i.e. that of a flower shop. While both accept a sender and receiver address, FreshFlowerShop service makes receiver address optional. It also provides a delivery receipt that is sent to the sender. Besides this, their descriptions use different terms for same concepts (e.g. FromAddress and SenderAddress).

Current matching tools are not likely to select FreshFlowerShop service as one of the matches if the user puts in a request for a service with same specification as that of FragrantFlowerShop service. The syntactic difference in their input specification can either be resolved through existing similarity based matching techniques [7] or through the use of an ontology that defines equivalence relationship between FromAddress and ToAddress, for example. The difference in their outputs and effects, however, would result into a mismatch with current matching techniques even though FreshFlowerShop service can be used to serve a request for FragrantFlowerShop service due to the existence of a semantic relationship between them [10].

Services in Registry

Name: FreshFlowerShop Service

Input: FromAddress, ToAddress, FlowerName, NumOfFlowers

Output: OrderReceipt, Packet, Amount, DeliveryReceipt

Precon: FromAddress **available**

Effect: OrderReceipt **sentTo** FromAddress, Amount **available**,

Packet **available**, DeliveryReceipt **sentTo** FromAddress **Name:** FragrantFlowerShop Service

Input: SenderAddress, ReceiverAddress, FlowerName, NumOfFlowers

Output: OrderReceipt, Packet, Amount

Precon: SenderAddress **available**, ReceiverAddress **available**,

Effect: OrderReceipt **sentTo** SenderAddress, Amount **available**, Packet **available**

Fig. 4. The FlowerShop Services

Specifically, FreshFlowerShop service is a *subtype* of FragrantFlowerShop service and can be used in its place. The representation of a Service as a combination of Service Type and Service Instance enables compile time matching of service interfaces. Software Developers can develop their service oriented programs using the Service Type descriptions available at compile time, as shown in Figure 3 (b). This program can later be instantiated after binding it with compatible Service Instances.

IV. SERVICE MATCHMAKING

Most of the existing approaches take a simplistic view of service matching. First, the semantic distance is computed for service attributes that are expressed merely as ontological concepts [8, 9] whereas actual descriptions could contain complex expressions as preconditions and effects of different operations. Second, the only service level operation available is *equivalence* that returns whether an exact or a lesser degree match exists between the services compared [12, 13]. Third, entire matching is performed at runtime introducing delays in the service discovery [14] and composition processes.

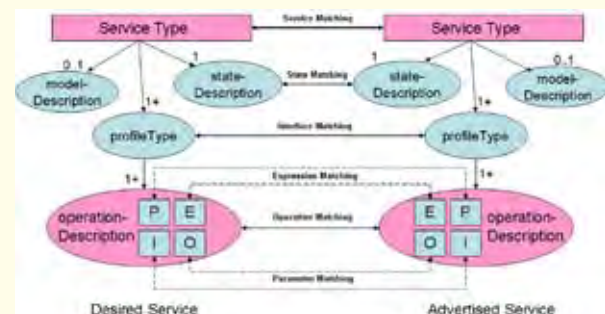


Fig. 5. Layered Matching

An end-to-end view of functional service matching is presented in Fig. 5. To match a desired service with an advertised service, the internal components of the two service descriptions need to be matched first. This happens at different levels of abstraction.

Parameter Matching: Here, matching is done to compare individual attributes (such as input elements or output elements) involved in service descriptions. As mentioned above, the attributes are typically ontological concepts and a similarity measure is defined that represents the semantic distance between two attributes.

Expression Matching: When ontological concepts alone are used to represent all kinds of preconditions and effects, it can lead to ontology explosion and also result in a brittle ontology [15]. Expression matching

defines a similarity measure representing semantic distance between two (boolean) expressions defined over ontological concepts.

Operation Matching: The operation level matching process uses parameter level matching results and expression level matching results to determine whether all $\langle I, O, P, E \rangle$ of the two operations being compared have a semantic match value above the specified threshold.

Interface, State and Model Matching: A service interface is a logical collection of related operations that the service offers. Therefore, interface level matching is simply a collection of operation matching results. State level matching determines the similarity of two services in terms of the internal state that they maintain. It is similar to parameter level matching since state is captured using simple ontological concepts. Matching based upon the internal process model of the services is called Model matching. It can be useful for temporal matching of services to ensure that services carry out certain steps in a particular order [16].

Service Matching: Similarity of two services is computed by aggregating the semantic distances between their corresponding operations and state descriptions.

CONCLUSION

In this paper, we presented a mechanism for specifying IT artifacts, to be managed, in a unifying representation based upon a semantic model for services.

This approach enables a new programming model for service oriented software development and enables automated discovery of desired or compatible services. Such functionality introduces elements of automated management since failed or non-complying artifacts (represented by services) can now be replaced by automatically searched alternatives. In future, we intend to continue this work and build a management system based upon techniques discussed in this paper.

REFERENCES

- [1] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web Services Architecture, W3C Working Group Note. <http://www.w3.org/TR/ws-arch/wsa.pdf>, Feb 2004.
- [2] Simple Object Access Protocol. <http://www.w3.org/TR/SOAP/>.
- [3] Web Services Description Language. <http://www.w3.org/TR/wsdl>.
- [4] OWL-S. <http://www.daml.org/services/owl-s/1.1/>, Nov 2004.
- [5] V. Agarwal, K. Dasgupta, N. Karnik, A. Kumar, A. Kundu, S. Mittal, and B. Srivastava. A Service Creation Environment based on End to End Composition of Web Services. In Proceedings of WWW, May 2005.
- [6] C. Facciorusso, S. Field, R. Hauser, Y. Hoffner, R. Humbel, R. Pawlitzek, W. Rjaibi, and C. Siminitz. A Web Services Matchmaking Engine for Web Services. In Proceedings of 4th Intl. Conf. on e-Commerce and Web Technologies, September 2003.
- [7] T. Syeda-Mahmood, G. Shah, R. Akkiraju, A. Ivan, and R. Goodwin. Searching Service Repositories by Combining Semantic and Ontological Matching. In IEEE International Conference on Web Services (ICWS), 200
- [8] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Semantic matching of web services capabilities. In Proceedings of the First Intl. Semantic Web Conference, pages 333–347, 2002.
- [9] P. Doshi, R. Goodwin, R. Akkiraju, and S. Roeder. Parameterized Semantic Matchmaking for Workflow Composition. Technical Report RC23133. <http://dali.ai.uic.edu/pdoshi/research/RC23133.html>, March 2004.
- [10] A. Kumar, A. Neogi, and D. J. Ram. An OO Based Semantic Model for Service Oriented Computing. In Proc. of IEEE SCC, USA, Sept. 2006.
- [11] V. D'Andrea, I. Fikoura, and M. Aiello. Interface Inheritance for Object-oriented Composition based on Model Driven Configuration. In *Proc. of ICSOC*, Nov 2004
- [12] N. Oldham, K. Verma, A. Sheth, and F. Hakimpour. SemanticWS-agreement partner selection. In Proceedings of the 15th Intl. Conference on World WideWeb, May 2006.
- [13] K. Verma, R. Akkiraju, and R. Goodwin. Semantic Matching of Web Service Policies. In Proc. of 2nd Intl. Workshop on Semantic and Dynamic Web Processes, 2005.
- [14] S. B. Mokhtar, A. Kaul, N. Georgantas, and V. Issarny. Towards Efficient Matching of Semantic Web Service Capabilities. In Intl. Workshop on Web Services Modeling and Testing, 2006.

- [15] A. Kumar, B. Srivastava, and S. Mittal. Information Modeling for End to End Composition of Semantic Web Services. In Proceedings of 4th Intl. Semantic Web Conference, Ireland, Nov 2005.
- [16] S. Agarwal and A. Ankolekar. Automatic matchmaking of web services. In Proceedings of 15th Intl. World Wide Web Conference, 2006.
- [17] A. Kumar and D. Janakiram. Towards a Programming Language for Services Computing. In Proceedings of the 17th International World Wide Web Conference (WWW), Beijing, China, April 2008.
- [18] A. Kumar, A. Neogi, S. Pragallapati, and D. J. Ram. Raising Programming Abstraction from Objects to Services. In Proceedings of IEEE Intl. Conference on Web Services (ICWS), Salt Lake City, USA, July 2007.
- [19] A. Kumar, S. Pragallapati, A. Neogi, and D. Janakiram. Raising Programming Abstraction from Objects to Services. In Proceedings of ICWS, July 2007.

Constructing a School Domain OWL Ontology Using Ontology Editor: A Case Study

B. Vinoth Raj¹, Sanjay Kumar Malik²

¹MCA (SE) VI Sem, University School of Information Technology, GGS Indraprastha University
Kashmere Gate, New Delhi, India

²University School of Information Technology, GGS Indraprastha University
Kashmere Gate, New Delhi, India

¹vinoth.amu@gmail.com, ²sdmalik@hotmail.com

Abstract — Semantic web has the potential to empower the current web to a level that will enable machines to comprehend the semantics (meaning) of web documents. Ontologies play a pivotal role in semantic web and can be used to capture knowledge about any domain of interest. The Web Ontology Language (OWL) is a semantic markup language for sharing ontologies on the web. OWL is designed for use by software agents to empower them to comprehend the meaning of web documents. Protégé is the most widely used ontology editor for developing ontologies and OWL is the most preferred ontology language by users. The paper presents an overview of how an OWL ontology can be constructed using Protégé 4.0 by integrating with an example of School domain.

Furthermore, the fundamental steps involved in the development of a domain ontology are identified and hence the construction of a class hierarchy (also called taxonomy) for the School domain is illustrated with the help of code snippets. Following that, object properties and members (individuals) are illustrated with the example considered.

INTRODUCTION

Current research and development in the realm of semantic web is likely to pave way for a web technology that will enable and empower machines to an extent that users' query will be answered more precisely. Tim Berners-Lee envisions semantic web as the future web. Ontologies provide a formal semantics that can be employed to process and integrate information on the web. A number of ontology editors are available for developing an ontology, e.g, Protégé, SWOOP, OntoEdit, Altova SemanticWorks, OntoStudio, and hence forth. But Protégé is most widely used by researchers, professionals, programmers, and others alike [1]. Also, OWL, a recommendation from W3C, is widely used to construct a domain ontology. Therefore, the paper focuses on how an OWL ontology is constructed using Protégé 4.0 version [2].

ONTOLOGY

Gruber [3] describes ontology as an explicit specification of conceptualization. Ontologies play a pivotal role in providing a vocabulary comprising unambiguous definitions for terms that can essentially serve as a formal support for communication between software agents. They provide a communication mechanism for users and software agents, clearly define the semantics for different domains for the purpose of interactions on the web, and help in creating a knowledge base that will enable people to work on a particular domain [4]. CYC upper ontology [5], LinkBase [6], and Ontolingua [7] are some of the well-known ontologies. A major reason for the recent increasing interest in ontologies is the development of the Semantic Web [7- 10].

Protégé is a free, open source ontology editor and knowledge base framework [11]. According to a survey conducted by Cardoso [1], Protégé editor and OWL ontology language are most widely used for the development of ontologies (see Fig. 1). Cardoso found that Protégé tool had a market share of 68.2% followed by Swoop, OntoEdit, Texteditor, Altova SemanticWorks, and hence forth. Also, Cardoso found that ontologies were mostly developed in the field of education (31%). Therefore, in this paper, the School domain ontology is constructed using Protégé and OWL.

CONSTRUCTING OWL USING PROTÉGÉ 4.0

Although numerous ontology editors are available today, the following fundamental steps are key to creation of a domain ontology in any development tool [12-15]:

- Obtain domain knowledge: A deep insight into and a thorough knowledge of the respective domain is prerequisite to construction of any domain ontology.
- Identify the key concepts: Concepts that represent the domain are identified and hence implemented by means of classes.
- Build the taxonomy: the class hierarchy is created by creating the classes and their respective subclasses, and instances of classes.

- Identify relationships between classes: Properties are used to represent relationship between classes.
- Consistency checking: the constructed domain ontology must be checked for consistency using reasoners.
- Implementation of ontology: involves deployment of ontology to enable machine-to-machine communication.

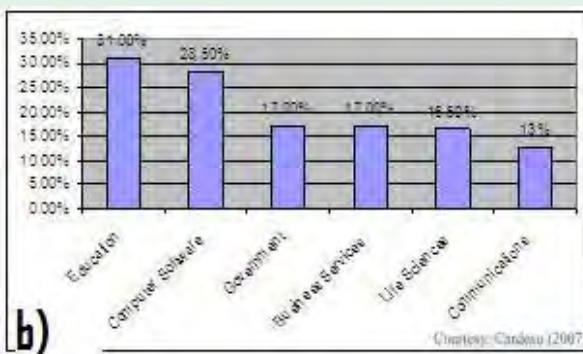
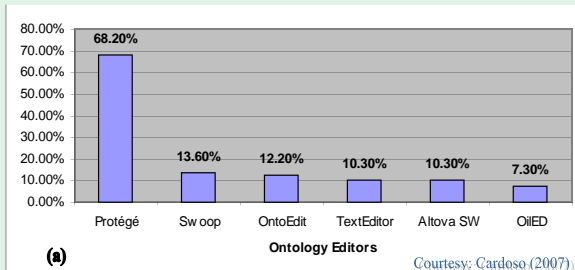


Figure 1. (a) Ontology editors used by respondents (researcher, professional, programmer, etc.). (b) Development of ontologies in different domains (Source: Jorge Cardoso, “The Semantic Web Vision: Where are We?” IEEE Intelligent Systems, September/October 2007, pp.22-26, 2007.).

CONSTRUCTING OWL USING PROTÉGÉ 4.0

Although numerous ontology editors are available today, the following fundamental steps are key to creation of a domain ontology in any development tool [15]:

- Obtain domain knowledge: A deep insight into and a thorough knowledge of the respective domain is prerequisite to construction of any domain ontology.
- Identify the key concepts: Concepts that represent the domain are identified and hence implemented by means of classes.
- Build the taxonomy: the class hierarchy is created by creating the classes and their respective subclasses, and instances of classes.
- Identify relationships between classes: Properties

are used to represent relationship between classes.

- Consistency checking: the constructed domain ontology must be checked for consistency using reasoners.
- Implementation of ontology: involves deployment of ontology to enable machine-to-machine communication.

OWL builds on RDF and RDF-S and uses RDF’s XML-based syntax. OWL documents are usually called OWL ontologies and are RDF documents [16]. The root element of OWL ontology is an `rdf:RDF` element, which specifies a number of namespaces as follows:

```
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2008/9/School.owl#"

```

```
xml:base="http://www.semanticweb.org/ontologies/2008/9/School.owl"

```

```
xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"

```

```
xmlns:School="http://www.semanticweb.org/ontologies/2008/9/School.owl#"

```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

```

```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

```

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```
xmlns:owl="http://www.w3.org/2002/07/owl#">

```

If an ontology is supposed to contain too many classes then creating each class and subclass can sometimes become a tedious process. Hence, Protégé 4.0 provides a facility called “Create Class Hierarchy” in the Tools option that can be used to do away with such repetitive tasks (see Fig. 2). Fig. 3 shows how subclasses are created. The OWL Properties construct represent relationship between classes. There are two types of object properties, namely object properties and datatype properties. Object properties are used to link individuals or members (see Fig. 4). To create an object property, go to the Object Properties tab and click Add Property button. Enter the object property name and create *teachesIn* and *hasTeacher* one by one as names of object properties. Now, select *teachesIn* and mention the Domains and Ranges by selecting the respective classes from the class tree. Now add the inverse property *hasTeacher*.

Fig. 5 illustrates how a datatype property can be created and also how a restriction can be added to a class. To create the datatype property *hasStrength*, switch to “Datatype Properties” tab and click the “Add Datatype Property” button. *hasStrength* is made functional so that any instance of Standard can only have only one particular value at one point of time. Now in the Classes tab, select the class Standard and click the “+” of equivalent properties. A new window will open as shown in Fig. 5. In order to specify a restriction that

the strength of a standard is greater than or equal to 20, we will select the “Class Expression Editor” and type the expression *Standard that hasStrength some int[>=20]*.

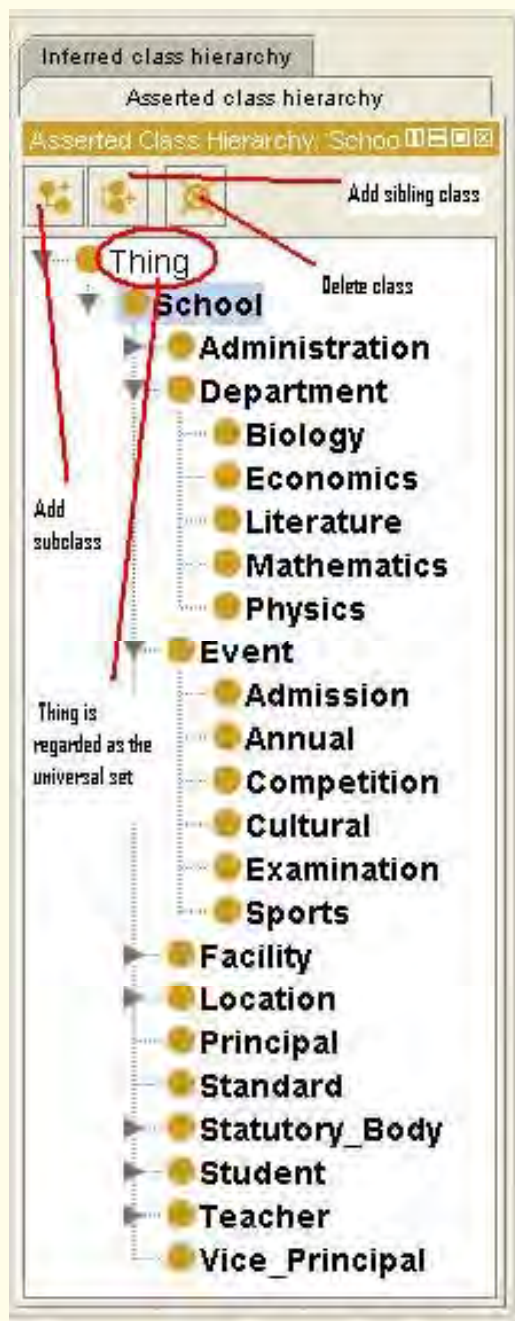
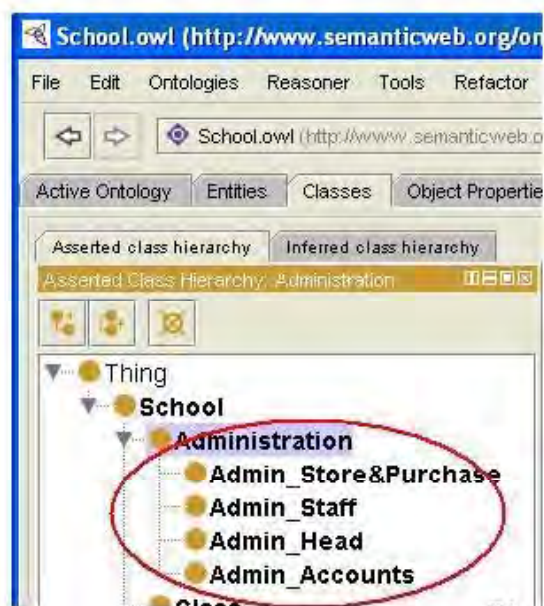


Figure 2. Class hierarchy (also called taxonomy) of the School domain.



(a)



(b)

Figure 3. (a) Adding prefix to the to-be-created subclasses. (b) Subclasses that have been created (encircled in red) using the Create Class Hierarchy facility.



Figure 4. The object property teachesIn has domain Teacher and range Standard.

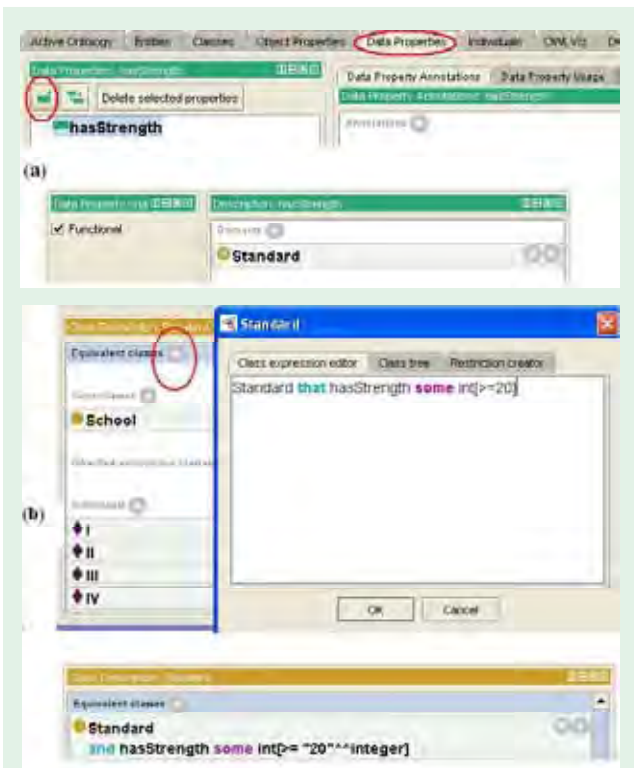


Figure 5. (a) Creating datatype property hasStrength. (b) Adding a restriction to the class Standard.

CONCLUSIONS AND FUTURE WORK

A key feature of ontologies is that, through formal, real-world semantics and consensual terminologies, they interweave human and machine understanding [17, 18]. Moreover, sharing and reuse of ontologies can empower machines to communicate the semantics of web documents easily. Semantic web can be a very powerful web technology that can revolutionize the information integration and retrieval process in a distributed system. Also, because Protégé tool and OWL are increasingly used by researchers and other experts alike, we have illustrated in this paper with the help of code snippets how the School domain ontology can be constructed using Protégé 4.0. This paper can guide in further research to construct an OWL ontology using Protégé 4.0. Furthermore, because a domain ontology requires consistent maintenance and development, this paper paves way for future research work in merging, debugging, and implementing OWL ontologies.

REFERENCES

- [1] Jorge Cardoso, "The Semantic Web Vision: Where are We?" *IEEE Intelligent Systems*, September/October 2007, pp.22-26, 2007.
- [2] Protégé 4.0 beta (build 102), Available at <http://protege.stanford.edu/download/protege4/installanywhere/>
- [3] T. R. Gruber. "A translation approach to portable ontology specifications", *Knowledge Acquisition*, 5:199–220, 1993.
- [4] T.S. Dillon, E. Chang, P. Wongthongthom, "Ontology-Based Software Engineering—Software Engineering 2.0", 19th Australian Conference on Software Engineering.
- [5] OpenCyc, Available at <http://www.opencyc.org/>
- [6] LinkBase, Available at <http://www.landcglobal.com/pages/linkbase.php>
- [7] Ontolingua, Available at <http://www.ksl.stanford.edu/software/ontolingua/>
- [8] Dieter Fensel et al., *Enabling Semantic Web Services: The Web Service Modeling Ontology*, Springer, 2007
- [9] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web", *Scientific American*, 284(5):34–43, May 2001.
- [10] Hai H. Wang, Natasha Noy, Alan Rector, Mark Musen, Timothy Redmond, Daniel Rubin, Samson Tu, Tania Tudorache, Nick Drummond, Matthew Horridge, Julian Seidenberg, "Frames and OWL Side by Side", 2006.
- [11] Protégé, Available at <http://protege.stanford.edu/>
- [12] W3C, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax, 2008. Accessed at <http://www.w3.org/TR/owl2-syntax/>
- [13] Choosing Between Versions of Protégé. Available at <http://protegewiki.stanford.edu/index.php/Protege4Migration>
- [14] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, Chris WroeA, "Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools", The University Of Manchester, October 16, 2007
- [15] Ontology Building: A Survey of Editing Tools, Available at <http://www.xml.com/pub/a/2002/11/06/ontologies.html?page=1>
- [16] Grigoris Antoniou and Frank van Harmelen, *A Semantic Web Primer*, The MIT Press, 2004
- [17] Jorge Cardoso, Martin Hepp, and Miltiadis Lytras (Eds.), *The Semantic Web: Real World Applications from Industry*, 2007.
- [18] D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd edition. Springer, Berlin/Heidelberg, 2003.

Application of Semantic Web in Agriculture Sector Giving everyone the 'Right to Internet access transforms to giving everyone the 'Right to Food!'

B. Indira Reddy

Principal, St.pauls P.G.college, R.R. Dist, Andhra Pradesh,India

Email: indira2259@yahoo.co.in

Abstract— In today's information age rapid technological changes are taking place at the global echelon. The most efficient World Wide Web which connects the world employ relatively simple technologies with sufficient scalability, efficiency and utility that they have resulted in a remarkable information freedom of organized resources growing across languages, cultures, and so on.. They are identification of resources, representation of resource state, and the practices that support the interaction between agents and resources in the space. Extending the web services with the term called semantic web for extended services and relating to the agriculture sector wherein it is considered the backbone of India's development and also to make an attempt to provide innovative methodologies with the help of ontology services as a tool for extending agricultural services for semantic web. This study will enable the agricultural research scientists to increase production, conserve the environment and so on and in turn helps the farmer to optimum incorporate in their day to day life.

Keywords— *Semantic Web, Ontology, Agrometeorology Resource Description Framework, Rice.*

I. INTRODUCTION

This is an attempt to study Web technologies, ranging from the basic technologies underlying the Web (URI, HTTP, HTML) to more advanced technologies being used in the context of Web engineering, for example structured data formats and Web programming frameworks. The goal of this study is to provide an overview of the technical issues surrounding the Web today, and to provide a solid and comprehensive perspective of the Web's constantly evolving landscape which resulted in the evolution of Semantic web(Tim berner'sLee dream).The scope of Semantic web is no limit, Extending the semantic web to agriculture sector to come up with results that can advance research and engender technologies that agricultural Scientists can make use of to increase production, conserve the environment and so on .The semantic web technology may be used to meet the following challenges:

1. Knowledge management challenges
2. Inefficient mechanisms and infrastructure for transferring technologies produced as the result of research to growers.
3. Keeping the indigenous knowledge as a heritage for new generations.
4. Easily accessing and availing economic and social knowledge.

This year marked yet another peak in rice production witnessed by the country with 99.37 million tones. The National Food Security Mission commissioned by the Govt. of India during 2007-08 had set the target of rice production to be raised by 10 million tons by the end of XI plan from the base line value of 92.76 million tones of production during 2006-07. The target now appears feasible. Illustrations (drawings, photographs, tables) should generally fit to one column (3.25" or 82mm). However, if they cannot be reduced to one column, place them across two columns at the top or bottom of the page.

DRR(Directorate of Rice Research) in its 45th year of useful existence has contributed significantly in overall rice production front which has ensured food security for the country. Even though the statistics look great, prices are shooting up and there are lakhs of farmers who can not fill their belly at least once a day because of crop loss due to sudden adverse effects of climate. However, rice production is greatly monsoon dependent. Any fluctuation in pattern and total rainfall from the normal will have adverse impact on the production front.

It is thus imperative to focus and reprioritize our research agenda to mitigate adverse effects of floods and drought on rice production. In a broader sence the Indian agriculture and it's economy are strongly influenced by the vagaries of the weather. The farming community is in great need to have access to weather information to plan and manage their crops and their livelihoods. Attempt is thus made to use the semantic web interface to provide valuable agromet information to the users . Internet technologies are already in use for dessiminating the information to the growers through Crop Weather Outlook of ICAR and several



World Population 6,830,853,300 Almost half of these people depend on rice. productive land in hectares 8,550,458,484, source: www.IRRI.org

other sites for a wider use by the planners, researchers, farming community and other public users. Despite its explosive growth over the last decade, the Web remains essentially a tool to allow humans to access information. The next generation of the Web, the 'Semantic Web', will extend the Web's capability through the increased availability machine-processable information. These machine-processable descriptions of Web information resources are called meta-data and are associated with ontologies, or conceptualisations of the domain of application. Meta-data and associated ontologies then allows more intelligent software systems to be written, automating the analysis and exploitation of Web-based information.

It also describes how knowledge management can be improved through the adoption of Semantic Web technology. To realize this, a number of different technologies need to be brought together. Their fusion provides the infrastructure which makes semantic knowledge management possible. Specifically, the use of knowledge discovery and human language technology to (semi-)automatically derive the required ontologies and meta-data, along with a methodology to support this process. The techniques describe for management and controlled evolution of ontologies and a set of semantic knowledge access tools for enhanced information access. Finally, a set of application scenarios for the technology are sketched.

The CWOL (Crop Weather Outlook) envisages to provide agrometeorological information highlights generated under the All India Coordinated Research Project on Agrometeorology (AICRPAM) and its Cooperating and Collaborating Centres along with 'Value Added Agro-advisory Reports'. Abundant work is done on Climate change and its impact, adaption and vulnerability on Indian Agriculture with the following objects.

- To identify the regions experiencing significant climate change and variability.
- To develop methodologies for assessing the impacts of climate change on agricultural productivity in various agro-ecological regions.
- To suggest suitable interventions for reducing the impacts of climate change on agricultural productivity.

II. REVIEW OF LITERATURE

The literature about the above research activities is made available on the world wide web which is rich of information and knowledge, makes huge amount of information available on the common platform, and helps to create knowledge-based society available in the form of static HTML pages. As literature says information and knowledge is made to be share among people, which increases its quality and usability, is becoming an important issue now a days. . but still in some developing countries like India where information technology is on emerging face, bulk of information restricted to hardcopy or stored in local databases that is either not open or not easy accessible to all. In India all the agricultural research institutes and universities are responsible to provide current information related to new innovations and better practices to farmer as well as research fellows. Usually agricultural organizations share their information through libraries and publications, which is mostly physically accessible, and very few of them are available on web and rest is moving towards digitization. But after all these efforts, still large sum of data which can be information and documents are left over in databases which are either not available as soft material or not uploaded on the website. In India every educational institute is gradually moving towards technology adaptation and currently all sort of information can be searched and archived through the web but some time search queries could not meet the appropriate result or come with no result because the information may not be available in form of document and text or not uploaded on the web and also the communication and information exchange between agriculture universities is also very poor due to lack of proper networking and connectivity facility.

III. SEMANTIC WEB APPLICATIONS:

The Semantic Web, with metadata annotated information, will be even more vital for completing information-based tasks On the Semantic Web, agents and other automated processes will produce more information faster and at a finer level of access and semantic granularity that can be shared via web services.

Attempting to manage this torrent of information using multiple applications will lead to a proliferation of applications, further partitioning related information thereby exacerbating the current complex, time consuming and error prone nature of many information-based tasks. A more robust solution is needed that can easily adapt to evolving user and task needs by working equally well with multiple, unanticipated types of information fragments as they become available on the Semantic Web. The Semantic Web hints at a solution to some of these problems. It offers a single unified data model powerful enough to hold all of the information currently scattered among multiple applications and the metadata annotations can be used to select relevant information. But merely unifying the data is insufficient. To use it to solve a particular task, users still need tools that will aggregate the information they need into a meaningful presentation that lets them view and manipulate it as is needed for their task. What are needed are small, flexible and reusable units of content and their associated user interfaces and application logic that can be arbitrarily combined to yield larger, more powerful task interfaces.

IV. CHALLENGES IN THE ENVIRONMENT:

Possible impacts of climate change on rice production are now being studied under controlled CO₂ enhanced open top chambers. Another landmark achievement of DRR during the year has been the release of four high yielding varieties viz., Improved Samba Mahsuri, Akshaya Dahan, Varadhan and Sampada by CSCSNRV. Consolidating on the gains offered by the hybrid rice technology, hybrids now occupy about 1.4 million ha. One of the recently identified hybrids, DRRH44 meets the long standing need for a good grain and cooking quality traits in hybrids. Marker assisted breeding efforts resulted in release of Improved Pusa Basmati and Improved Smaba Mahsuri with high level of BLB resistance, while extensive testing of Swarna-sub1 and IR64-sub1 reposed confidence in these submergence tolerant varieties. Use of ICTs and GIS based tools in technology transfer is another new area of focus covered in the report.

There has been significant interest in applying the practices of semantic web to build an online repository of agricultural information in recent times. An ongoing project titled "Re-designing the farmer-extension-agricultural research/ education continuum in India with ICT-mediated knowledge management" has created AGROPEDIA, pioneered by experts from IIT-K which is immense use to the agro scientists.

CONCLUSIONS

In this endeavour, an in-depth analysis of the Strengths, Weaknesses, Opportunities and Threats (SWOT) was undertaken to place research and technology development efforts in perspective so that we succeed in our pursuit of doing better than the best. With the application of semantic web and ontology based information Indian agriculture must continuously evolve to remain ever responsive to manage the change and to meet the growing and diversified needs of different stakeholders in the entire production to consumption chain.

This is an attempt to visualize an alternate agricultural scenario from present to twenty years.

REFERENCES

1. The Semantic web an Introduction
2. U.N.'s FAO Aims At Agricultural Services Based On Semantics – Across Multiple Languages
3. Semantic web application areas.
4. Center for
5. The Semantic Web: Real World Applications from ... - Jorge Cardoso, Martin Hepp, ... - 2008 - 316 pages
6. Introduction to the Semantic Web and Semantic ... - Liyang Yu - 2007 - 368 pages
7. Ontology Learning for the Semantic Web - Alexander Maedche - 2002 - 280 pages
8. Agricultural Ontology Service - Wikipedia, the free encyclopedia
9. Research on the Semantic Web-based Technology of Knowledge doi.ieeecomputersociety.org/10.1109/FSKD.2009.58
10. Directorate of rice research, Hyderabad
11. ARI Agricultural Research Institute, Hyderabad
12. ICRISAT, Hyderabad
13. U.N.'s FAO Aims At Agricultural Services Based On Semantics www.semanticweb.com/.../unas_fao_aims_at_agricultural_services_based_on_semantics
14. Semantic Web: Revolutionizing Knowledge ... - Christopher JO Baker, Kei Hoi Cheung – 2007
15. Spinning the Semantic Web: Bringing the World ... - Dieter Fensel, James A Hendler, Henry ... – 2005
16. **Use of Semantic web and ontology for agriculture education system**
vasatwiki.icrisat.org/.../Use_of_Semantic_web_and_ontology_for_agriculture_education_system_in_India
17. *RICE IS LIFE-DRR* www.drricar.org/
18. <http://agropedia.net/>

Programming API to create RDF statements

Dominik Tomaszuk, M.Sc¹

¹University of Bialystok, Poland
dtomaszuk@ii.uwb.edu.pl

Abstract — This paper describes an Application Programming Interface (API) for creating Resource Description Framework (RDF) triples. It propose a standard in a general, programming language independent way to manage RDF data. This paper defines interfaces, methods and attributes using Interface Definition Language (IDL). This new API is developed in response to a demand from a wide range of users, who want create RDF statements.

Keywords — Semantic Web, Resource Description Framework (RDF), Interface Definition Language (IDL), Application Programming Interface (API), Object-oriented programming (OOP).

I. INTRODUCTION

A simplified view of the Semantic Web is a collection of web retrievable RDF documents, each containing a Resource Description Framework (RDF) graphs. RDF is designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources. In other words RDF is a general-purpose language for representing information in the Web.

RDF Recommendations [1, 2, 3, 4] explains the meaning of subject, predicate and object. These expressions are known as triples in RDF terminology. The subject denotes the resource. The predicate means traits or aspects of the resource and expresses a relationship between the subject and the object. A collection of RDF statements intrinsically represents the labeled, directed multi-graph.



Fig. 1. RDF graph data model

More formally, let U to be the set of all URI references, B an infinite set of blank nodes, L the set of RDF plain literals, and D the set of all RDF typed literals. All the four sets are defined in [1]. U , B and L are pairwise disjoint. Let $O = U \cup B \cup L \cup D$ and $S = U \cup B$, then $T = S \cup O$ is set of all RDF triples¹.

There are not any standard programming API to manage RDF data. But there are many libraries that provide various methods and attributes to create RDF triples..

II. INTERFACE DEFINITION LANGUAGE BASICS

The Interface Definition Language (IDL) [5] is a specification language used to describe a software component's interface. An interface definition written in IDL completely defines the interface and fully specifies each operation's parameters. IDL describes an interface in a language-neutral way and programming language independent.

The IDL is produced by the Object Management Group (OMG). It is a declarative language, which primarily describes an object-oriented languages. OMG has defined mappings from IDL to just about every major programming language, such as: C, C++, Java, Smalltalk, Ada and Python.

To declare types IDL use interfaces delimited by curly braces. Programmers can work with structs, sequences, attributes, valuetypes, and other IDL types and constructs to create very flexible data structures for their interfaces. Listing 1 presents an example of IDL interface.

```

interface calculator {
    float subtract ( in float minuend, in float
        subtrahend );
}

```

Listing 1. Simple IDL interface.

III.IDL DEFINITIONS OF BASIC DATATYPES

It is proposed to introduce a new, universal programming API standard to create RDF triples. It is proposed a hierarchy starting with `RDFTerm` with its descendants: `RDFLiteral`, `RDFPlainLiteral`, `RDFTypedLiteral`, `RDFResource`, `RDFURIReference` and `RDFBlankNode`. Fig. 1 presents the hierarchy in UML [6].

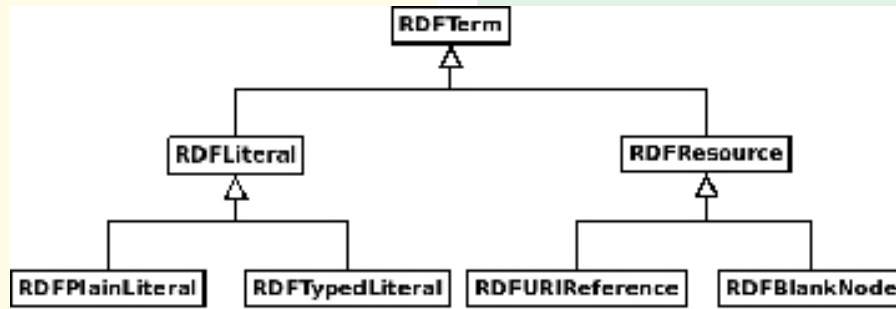


Fig. 2. UML Class diagram presents hierarchy of interfaces

IV. RDFS TERM INTERFACE

An RDFS Term is the abstract root type in the hierarchy of datatypes used in the RDFS API. This interface contains methods to create URI reference, blank node and plain literal or typed literal. Listing 2 presents RDFS Term interface.

```

interface RDFS Term {
    RDFS Literal createLiteral (in string value, in [optional]
    string language);
    RDFS Literal createLiteral (in string value, in RDFS URIReference
    type);
    RDFS URIReference createURI (in String value);
    RDFS BlankNode createBlankNode (in String id);
};
    
```

Listing 2. RDFS Term interface

The first createLiteral method creates a new RDFS Literal that is a plain literal in RDFS. The method has two parameters:

- value,
- language.

The value parameter is a string type. It is the lexical value of this literal. The language parameter is a type string and it is optional. It is the language tag defined in []. The createLiteral method returns RDFS Literal. The RDFS Literal is described as plain literal.

The second createLiteral method creates a new RDFS Literal that is a typed literal in RDFS. The method has two parameters:

- value,
- type.

The value parameter is a string type. It is the lexical value of this literal. The type parameter is URI type. It is the datatype identified by a URI reference. The createLiteral method returns RDFS Literal. The RDFS Literal is described as typed literal.

The createURI method creates a new URI

reference. The method has value parameter. The value parameter is a RDFS URIReference type. It is the lexical value of this URI. The createURI method returns RDFS URIReference.

The createBlankNode method creates a new blank node. The method has id parameter. The id parameter is a type string. It is the identifier of the blank node. The createBlankNode method returns RDFS BlankNode.

RDFS Literal, RDFS PlainLiteral and RDFS TypedLiteral interfaces

An RDFS Literal is an abstract type of RDFS PlainLiteral and RDFS TypedLiteral. Listing 3 presents RDFS Literal interface.

```

interface RDFS Literal : RDFS Term {
};
    
```

Listing 3. RDFS Literal interface

Listing 4 presents RDFS PlainLiteral interface.

```

interface RDFS PlainLiteral : RDFS Literal {
    readonly attribute string value;
    readonly attribute string language;
};
    
```

Listing 4. RDFS PlainLiteral interface

The value attribute is the lexical value of this literal. It is a string type. The language attribute is the two characters long language tag as defined by []. It is also a string type.

Listing 5 presents RDFS TypedLiteral interface.

```

interface RDFS TypedLiteral : RDFS Literal {
    readonly attribute string value;
    readonly attribute RDFS URIReference type;
};
    
```

Listing 5. RDFS TypedLiteral interface

The value attribute is the lexical value of this

literal. It is a string type. The type attribute is the datatype identified by an URI reference. It is an RDFURIRreference type.

RDFResource, RDFURIRreference, RDFBlankNode interfaces

An RDFResource is an abstract type of RDFURIRreference, RDFBlankNode. Listing 6 presents RDFResource interface.

```
interface RDFResource : RDFTerm {
};
```

Listing 6. RDFResource interface

Listing 7 presents RDFURIRreference interface.

```
interface RDFURIRreference : RDFResource {
    readonly attribute string value;
};
```

Listing 7. RDFURIRreference interface

The value attribute is the lexical representation of the URI reference. It is a string type.

Listing 8 presents RDFBlankNode interface.

```
interface RDFBlankNode : RDFResource {
    readonly attribute string id;
};
```

Listing 8. RDFBlankNode interface

The id attribute is identifier of the blank node. It is a string type.

V. IDL DEFINITION OF INTERFACE TO CREATE RDF STATEMENT

An RDFStatement defines the data structure to represent an RDF triples. Listing 9 presents RDFStatement interface.

```
interface RDFStatement {
    readonly attribute RDFResource subject;
    readonly attribute RDFURIRreference predicate;
    readonly attribute RDFTerm object;
    RDFStatement createStatement (in RDFResource
    subject, in RDFURIRreference predicate, in RDFTerm
    object);
};
```

Listing 9. RDFStatement interface

The RDFStatement interface has three attributes:

- subject,
- predicate,
- object.

The subject attribute is subject resource of the RDFStatement. This attribute is a RDFResource type. The predicate attribute is predicate URI reference of the RDFStatement. This attribute is a RDFURIRreference type. The object attribute is object term of the RDFStatement. This attribute is a RDFTerm type.

The createStatement method creates a new RDFStatement object. The method has three input parameters: subject, predicate and object.

CONCLUSIONS

I suggest that it is time that the Semantic Web community support a simple programming API such as mine. I believe my API is an interesting approach. Advantage of this API is a standard in a general, programming language independent way to manage RDF data. The proposal can be implemented in programming languages.

I realized that some further work on this issue is still necessary extended proposed API to iterating through RDF statements and querying for triples.

ACKNOWLEDGEMENTS

The author would like to thank Ivan Herman from the World Wide Web Consortium. Thanks to the W3C SPARQL Working Group for their comments on this work as it has developed.

REFERENCES

- [1] G. Klyne and J. J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium, 2004.
- [2] P. Hayes. *RDF Semantics*. World Wide Web Consortium, 2004.
- [3] F. Manola E. and Miller. *RDF Primer*. World Wide Web Consortium, 2004.
- [4] D. Brickley and R. V. Guha. *RDF Vocabulary Description Language 1.0*. World Wide Web Consortium, 2004.
- [5] L. Heaton, *OMG IDL Syntax and Semantics*, Object Management Group, 2002.
- [6] G. Booch, I. Jacobson, J. Rumbaugh, *Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2*, International Organization for Standardization (ISO/IEC 19501), 2005

Impact of Semantic Web Technology on Higher and Future Education

Dr. Mamta Malik¹, Deepak Yadav², Amma Naningrum³ and Dr. Anil Jain⁴

¹Lecturer, S.S.in Lib & Inf. Sc., Vikram University Ujjain (M.P.) drmmalik_ujn@yahoo.co.in

²Student, Dept of Journalism and Library and information Sc., Oslo University College, Norway
depuvadavind0@gmail.com

³Student, Dept of Journalism and Library and information Sc., Oslo University College, Norway
ammananingru@gmail.com

⁴Reader, S.S. in Lib & Inf. Sc., Vikram University Ujjain (M.P.) Ajk2011@rediffmail.com

Abstract— This article discusses the current state of the Semantic Web, and how it may impact on Higher and Further Education sectors over the next few years. It discusses the technology and tools now available to support it. The impact of the Semantic Web is likely to be particularly strong in four clear areas where there could be major implications for both teaching and research: in information management; in digital libraries; in support for interaction between virtual communities and collaborations; and in e-learning methods and tools. The Semantic Web clearly has large application to e-learning, supporting both distance and local education. It gives an overview on Academic Semantic web system and services to enhance the accessibility of Academicians materials as well. Lastly conclusion is drawn.

Key words— Semantic Web, Digital Library, E-Learning.

1. INTRODUCTION

The World-Wide Web is an important learning technology platform today. Its accessibility has made it a successful environment in particular for the publication of learning material. Learning resources can be provided in a standardised format that can be accessed at any time from any location. The Web, however, is still evolving. The current evolution of the Web can have an impact on educational technology. This will affect instructors and learners alike [1]. The Semantic Web initiative aims to support explicit semantics and its automated processing [2]. Currently, search and retrieval functionality relies on human interaction and often ad-hoc approaches to selection of documents for a given set of search criteria. Semantic annotations, which can be processed by software applications, will improve the precision of searches. This will enable accurate searches for learning resources. The opportunities that will emerge for educational technology as a result of the Semantic Web initiative, however, go beyond search and retrieval [3]. The overall development and deployment process of educational technology can be affected.

Ontology technology – the knowledge representation and inference core of the Semantic Web – promises this wide applicability [4]. An area such as education, where access to information is central, depends on the representation and organisation of knowledge both for the content but also the metadata level.

2. SEMANTIC WEB

The Semantic Web is an idea developed initially by Tim Berners-Lee (the inventor of the Web) in 1998 at the World Wide Web (W3C) Consortium and is defined on that Web site (www.w3.org/2001/sw/Activity) as having the goal of being a universal medium for the exchange of data [5]. The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [6]. The semantic web, also known as the internet of meanings, is not just a vision of the future [7]. The next step is to implement this new vision of the web as real-world applications [8]. The main goal of Semantic Web is to develop languages for expressing information in a machine processable way [9]. The Semantic Web is a proposed extension of the existing web, where information found on the web is augmented with machine-accessible knowledge [10]. The key enabler of the Semantic Web is the need of many communities to put machine-understandable data on the Web which can be shared and processed by automated tools as well as by people. Machines should not just be able to display data, but rather be able to use it for automation, integration and reuse across various applications. The European Commission is funding numerous projects related to Ontologies and the Semantic Web in its currently running Sixth Framework Research Programme, e.g. the SEKT project [11]. (“Semantically Enabled Knowledge Technologies”).

3. IMPACT OF SEMANTIC WEB ON HIGHER EDUCATION AND FUTURE EDUCATION

There are four clear areas where there could be major implications for both teaching and research:

in information management; in digital libraries; in support for interaction between virtual communities and collaborations; and in e-learning methods and tools.

3.1 Information Management and Discovery Tools

Perhaps the most widely developed space at the moment within the Semantic Web is in information management, i.e. the organisation and discovery of information. The Semantic Web enhances the capabilities of those tools which form a familiar part of the current Web so that they can become useful information management tools in their own right. The Web is already an information source of choice for many learners and researchers. A more structured and directed approach to managing this information space, both within institutions and across the whole community, can make this information more useful, with less wasted effort, and more capacity to measure the quality of information. By making the annotation machine readable, it becomes accessible to automatic processing, carrying out many routine tasks which consume people's time.

3.2 Semantic Web and Digital Libraries

Libraries are a key component of the information infrastructure which underpins higher and future Education. They provide an essential resource for students and researchers for reference and for research. And they are increasingly converting themselves to *Digital Libraries*. A key aspect for the Digital Library is the provision of *shared catalogues* which can be published and browsed. This requires the use of common metadata to describe the fields of the catalogue (such as author, title, date, publisher), and common *controlled vocabularies* to allow subject identifiers to be assigned to publications.

Metadata: it is a key component of the provision of online catalogues that are searchable across the Web. In order to use the Semantic Web to its best effect, metadata needs to be published in RDF formats. There are several initiatives involved with defining metadata standards in the library and publishing community, including:

- **Dublin Core Metadata Initiative** which provides a standard set of machine readable fields and guidelines for their use. This now has a well-established RDF vocabulary [12].
- **MARC.** The well known MARC format from the Library of Congress has an XML representation
- **ONIX.** The ONIX for Books Product Information Message is the international standard for

representing and communicating book industry product information in electronic form XML representation.

- **PRISM.** The Publishing Requirements for Industry Standard Metadata specification defines an XML metadata vocabulary for magazine, news, catalogue, book, and journal content [13].

Such standards can be used across the Web in that they provide a common metadata vocabulary in XML or RDF which can be used to mark up and share library catalogues on the Web.

Controlled Vocabulary Controlled vocabularies such as classifications, taxonomies and thesauri are the other key components for cataloguing and searching by classifying documents by subject. Developing tools and formats for representing and delivering such thesauri on the Semantic Web has been a major initiative of the SWAD-Europe project [14].

3.3 Supporting interaction

A major theme that has emerged during the development of the Semantic Web is the ability to support interaction between groups of people across the Web. This has two aspects: support for virtual communities and support for virtual organisations.

Semantic Web in Virtual Communities

Within virtual communities individuals can publish information about themselves, their interests and their work, and allow other like-minded individuals to discover and share that information in order to build a virtual community of people sharing ideas.

The 'Friend of a Friend' or FOAF [15] project provides a simple language that allows people to publish information about themselves, their work and interests, along with their contact details (with due respect to privacy). This is useful, but becomes interesting when people can also publish links to others they know in the community.

Semantic Web in Virtual Organisations

A more rigorous approach is being taken when people and organisations wish to formally collaborate towards common goals across the computer infrastructure. Examples of this in the higher education and future education community would include research projects with partners, or in computer-assisted learning, where students and teacher wish to share online teaching and learning resources, and engage in group activity such as a team project.

3.2 E-Learning

The Semantic Web clearly has large application to e-learning, supporting both distance and local education. The notion of a 'learning object' as a separable unit of educational material which can be reused and combined with other learning objects has been a central feature of e-learning systems. However, used properly, it is a useful and powerful concept and one which the Semantic Web has much to offer. Learning objects can be organised into repositories, and shared across peer-to-peer (P2P) networks.

4. ACADEMIC SEMANTIC WEB SYSTEM: AN OVERVIEW

Figure 1 shows the distributed architecture of academic semantic web. In this scenario, each organisation or higher educational institute will generate and maintain its own academic Ontology [16]. Thus, the academic knowledge is shareable over the academic Semantic Web through Semantic Web Services as stated next:

Scholarly service provider: This is a Web Service for scholarly information retrieval. It can retrieve appropriate knowledge stored in the academic Ontology from different organisations or educational institute. Services include finding relevant documents or locating experts in a certain research area.

Scholarly service requester: It is also a Web Service, which interacts with users for supporting scholarly query requests.

Matchmaking agent: It searches for the appropriate Scholarly Service Providers that can match the query specifications requested by scholarly service requesters. Then, it establishes the necessary network connections that connect the scholarly service requester and the scholarly service provider for supporting the query requests directly.

5. CONCLUSION:

The Semantic Web has great potential, and with direct application to the higher education and future education sector. However, it has been a long time in development and does require an investment of time, expertise and resources. Institutional libraries should be considering joining collaborations to explore how Semantic Web can best be exploited and investing in training staff, with a view to providing Semantic Web solutions within the next two to three years. Information science professionals and academics working in

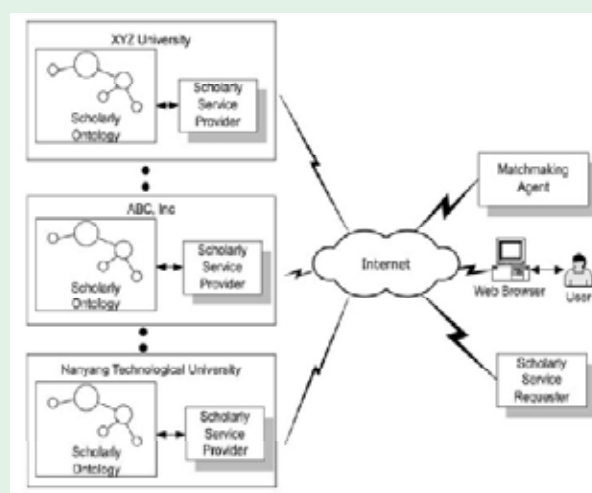


Fig. 1. Academic semantic web system (Source: Tho,Q.T., Fong, A.C.M. and Hui, S.C.(2006) A scholarly semantic web system for advanced search functions.

particular fields should work together to provide the vocabularies and domain ontologies required to support particular fields. Particular communities and research groups could be looking at exploiting the emerging infrastructure to enhance the interaction of their community. What we will have is a richer experience of IT that is better able to deliver the right information at the right time in the right way, so we can get on with the serious business of research and teaching.

REFERENCES

1. Pahl, C. and Holohan, E. (2005). Applications of semantic web technology to support learning content development. *Interdisciplinary Journal of E-Learning and Learning Objects*, Vol-5.
2. W3C – the World Wide Web Consortium. (2006). *The semantic web initiative*, available at: <http://www.w3.org/2001/sw>
3. Devedžić, V. (2004). Education and the semantic web. *International Journal of Artificial Intelligence in Education (IJAIED)*, 14, 165-191.
4. Henze, N., Dolog, P., & Nejdl, W. (2004). Reasoning and ontologies for personalised e-learning in the semantic web. *Educational Technology & Society*, 7(4), 82-97.
5. Goossens, P. (2003). Short communication ELAG 2002: report of a library systems seminar on the semantic web. *Program: electronic library and information systems*, 37 (4). 251-253. available at: <http://www.emeraldinsight.com/0033-0337.htm>
6. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 2001(5), available at <http://www.sciam.com/2001/0501issue/0501berners-lee.html>

7. Berners-Lee, T. (2000), "Semantic web", available at: www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html
8. W3C (2002), Semantic Web Activity Statement, available at: www.w3.org/2001/sw/Activity
9. Dumbill, E. (2001), The Semantic Web: A Primer, XML.com, 01 November, available at: www.xml.com/pub/a/2000/11/01/semanticWeb/index.html
10. Berners-Lee, T., Hendler, J. and Lassila, O. (2001), "The Semantic Web", Scientific American, Vol. 284 No. 5, pp. 34-43, available at: www.sciam.com/article.cfm?articleID=0004814410D2-1C70-84A9809EC588EF21
11. EU IST SEKT project, available at: <http://www.sekt-project.com>.
12. Dublin Core Metadata Initiative homepage: <http://dublincore.org>
13. Publishing Requirements for Industry Standard Metadata (PRISM) homepage: <http://www.prismstandard.org/about/>
14. SWAD-Europe Thesaurus Activity homepage: <http://www.w3.org/2001/sw/Europe/reports/thes/>
15. Friend of a Friend Project homepage: <http://www.foaf-project.org/>
16. Tho, Q.T., Hui, S.C., Fong, A.C.M. and Cao, T.H. (2006), "Automatic fuzzy ontology generation for semantic web", IEEE Trans. Knowledge and Data Engineering, Vol. 18 No. 6, pp. 842-56.

Semantic Web Architecture

Golwal Madansing D.#, Chavan Vishakha D.#

Research Scholar#, Dept. of Library & Information Science

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad – 431003 [Maharashtra]

mgolwal4@gmail.com

vishakha.sprit@gmail.com

Abstract - Semantic web is also popularly known as web 3.0 or web of data. Data and not documents are the building blocks of the semantic web. Although a lot of information has become readily accessible and necessary for daily work, the current infrastructure for managing information is ill-suited for information-oriented activities: information and functionality is scattered across applications and websites, making it difficult to aggregate and reuse just the right set of content and operations required for unique user tasks. In this paper we discuss a collection of tools built into the Haystack information management platform that address many of the shortcomings of current applications, and allow composing reusable fragments of information from the Semantic Web and the operations that manipulate them into a task workspace tailored to the user and the task.

Keywords – Semantic Web, Web 3.0, RDF, Metadata, Semantic Web Architecture, RDF, W3C.

I. INTRODUCTION

The concept of Semantic Web was introduced by Tim Berners-Lee, the developer of HTML, Hyper Text Transfer Protocol (HTTP), Uniform Resource Identifiers (URI) and World Wide Web (WWW). His visualization of Semantic Web is that in future we will have intelligent software agents that will analyze a particular given situation and present us with the best possible alternatives. In other words, the connectivity that is found today only on PCs through the Web, will become a part of our daily life.

The semantic web is a vision of the next generation web, which enables web applications to automatically collect web documents from diverse sources, integrate and process information and interoperate with other applications in order to execute sophisticated tasks for humans.

The idea behind Semantic Web is to develop such technologies that make the information more meaningful for the machine to process, which in turn makes search and retrieval of information more effective for humans. For instance, available Web technologies

include parsers which can validate the display of Web documents by checking for syntactical errors. But as of now, computers are unable to understand the semantics underlying the documents. For example, a computer cannot understand that a particular Web page is the homepage of an Institute or that of an individual; or that a hyperlink leads to the *resume* of a person.

II. SEMANTIC WEB

Semantic web is also popularly known as web 3.0 or web of data. The conception of Semantic Web is characterized by developing languages, tools, etc. that make information processing semantically by machines. And also a very important aspect of Semantic Web is development of standards and protocols, as there is hardly any consensus among the people working on projects about what the future Semantic Web will be.

The semantic web approach aims to develop languages for expressing information in a machine processable way. Tim Berners-Lee, who is the inventor of the World Wide Web, first envisioned a semantic web that provides automated information access based on machine-processable semantics of data. The explicit representation of the semantics of data, accompanied with domain theories (i.e. ontologies), will enable a web that provides a qualitatively new level of service.

Define

- The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.
- In the term “semantic web”, ‘semantic’ also indicates that the meaning of data on the web can be discovered- not just by people, but also by computers.

The semantic web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications.

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

III. ARCHITECTURE

The current architecture of the Semantic Web is based on the well-known Semantic Web stack described by Tim Berners-Lee in 2000. This is only a high-level picture of the Semantic Web, and thus leaves out a lot of details. The general impact of this picture, particularly as it has been interpreted during the development of RDF, RDFS, and OWL, is that RDF forms the basis of the Semantic Web, both for syntax and semantics. All Semantic Web documents thus should have the syntax of RDF, and this syntax should be read as encoding RDF triples which form the abstract syntax of RDF. Further, the meaning of these triples should include their RDF model-theoretic meaning that is, all triples can be thought of as atomic facts.

There are other aspects of the Semantic Web Architecture. These include the use of URI references as identifiers, XML Schema data types as data types, and the use of model-theoretic entailment as the primary semantic relationship. As well, the Semantic Web was envisioned as a stack of languages, each building directly and completely on the lower languages. Thus the ontology layer built on the RDF layer, and the logic layer built on the ontology layer. Recent accounts of the Semantic Web architecture have split the single stack into two side-by-side extensions of RDF for ontologies and rules. However, this does not change the fundamental role of RDF in the Semantic Web architecture.

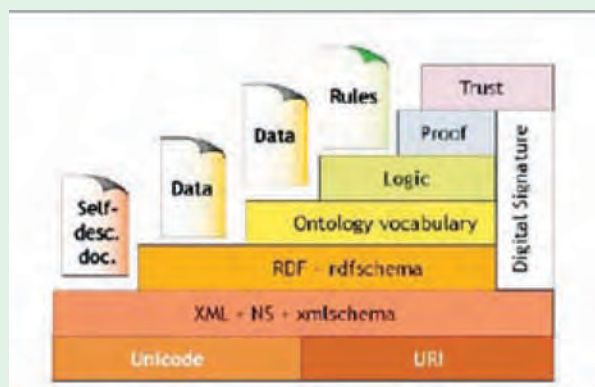


Fig. 1. Semantic Web Layer Approach

A. Layered Approach to Semantic Web : In building the semantic web in a layered manner, two principles should be followed:

- **Downward Compatibility:** Agents (Agents are pieces of software that work autonomously and proactively.) fully aware of one layer should also be able to interpret and use information written at lower levels. e.g. agents aware of the semantics of OWL can take full advantage of information written in RDF and RDF Schema.
- **Upward Partial Understanding:** agents fully aware of one layer should also be able to take at least partial advantage of information at higher levels. e.g. an agent aware of only RDF and RDF Schema semantics can interpret partial knowledge written in OWL, by disregarding those elements that go beyond RDF and RDF Schema.

The “layer cake” of the semantic web technology as shown in the above figure, describes the main layers of the semantic web design and vision.

B. The third common use of the term Semantic Web is to identify a set of technologies, tools and standards which form the basic building blocks of a system that could support the vision of a Web imbued with meaning. The Semantic Web has been developing a layered architecture, which is often represented using a diagram first proposed by Tim Berners-Lee, with many variations since. Figure 1 gives a typical representation of this diagram.

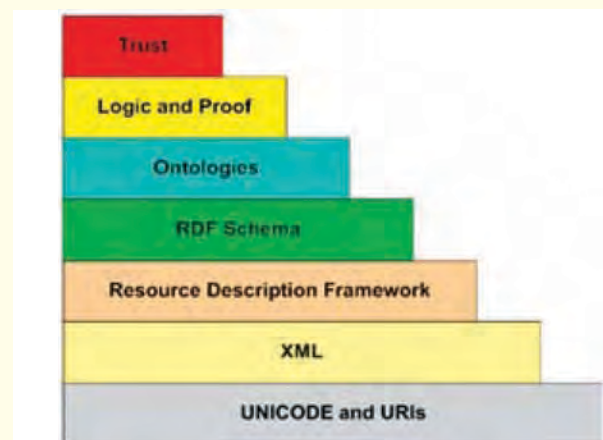


Fig. 2. Semantic Web Layer

- **Unicode and URI:** Unicode, the standard for computer character representation, and URIs, the standard for identifying and locating resources (such as pages on the Web), provide a baseline for representing characters used in most of the languages in the world, and for identifying resources.

- ## IV. SEMANTIC WEB ARCHITECTURE

- **Web Service Provider**

semantic web service registry.

Common Vocabulary (CV) registry shall provide an interface to submit new RDF/OWL ontology IRI. Registry shall also provide an interface to find context/synonym based normative metadata vocabulary¹ for a „thing“.

Semantic web service registry shall provide discovery service or an interface to submit IRI for semantically annotated WSD document. An interface shall be provided to find semantic web service WSD document for a specific vocabulary support.

- **Web Service Requester**

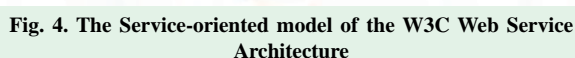
Semantic web browser shall have separate options to load HTML/XHTML/RDFa content („GO) and RDF ontology („S) from IRI. The browser may have options for graphical or tabular display of RDF ontology. To enable the display option „S HTTP:Accept application/rdf+xml media type shall be supported by the web browser.

Semantic search application shall scan the published web content and web services WSD documents for support of standard ontology. Semantic broker is a search engine (SE) built-in application [6]. SE shall build provider IRI database for supported metadata & vocabulary. SE shall find new standard ontology and semantic web services from respective registries to update its cache. SE may require GRDDL transformation of the semantic web content to update its cache; the transformation also updates hosting web server cache.

- **Infrastructure Elements**

Common Vocabulary (CV) registry shall validate & integrate new ontology and scattered vocabularies into existing vocabularies and approve new vocabularies; the validation & integration will eliminate redundant ontology.

Above figure shows the service-oriented model of the WSA. We now give a brief overview of it. The concept of a service is core, which is defined as an abstract resource representing a capability to perform some coherent set of tasks. The performance of a given task involves some exchange of messages between the requestor and provider agents. Choreography defines a pattern of possible message exchanges (i.e. conversations). The WSA associates choreography with



As Sycara et al observe, tasks can be defined in three ways. They can be represented explicitly, using name labels with a well defined semantics in some ontology of tasks. They can be represented implicitly, using a language of preconditions and effects of a task. Or they can be defined using a combination of these two approaches. The WSA makes no commitment as to which approach is used. The primitive operation within tasks is referred to in the WSA as an action. However, the authors observe that ‘the actions performed are largely out-of-scope... but the resulting messages are in scope.’ A service has a description, which specifies its interface and messages in a machine-processable way. This may also include a description of the service’s semantics. The semantics specifies (formally, informally or implicitly) the intended effect of using the service: specifically, the tasks that constitute it. The WSA views this semantics as ‘a contract between the requestor entity and provider entity concerning the effects and requirements pertaining to the use of a service.’

V. CONCLUSION

The objective of the proposed semantic web architecture is to bring order and control on the growth of vocabularies and facilitate discovery of standard ontology and semantic web services. The enforcement of this architecture will accelerate the acceptance of semantic technologies by web service providers and requesters. Consumer trust is established with demonstration of commercial advantage of semantic technologies. This architecture addresses the Trust (openness), Proof (correctness) & Logic (semantic technology) requirements as depicted in.

REFERENCE:

- 1) Anderson, C. and Horvitz, E. Web Montage: A Dynamic Personalized Start Page. Proceedings of the Eleventh International Conference on the World Wide Web, 2002.
- 2) Berners-Lee, T., Hendler, J. and Lassila, O. The Semantic Web. *Scientific American*, May 2001. <http://xwinman.org/>
- 3) Hogue, A. and Karger, D. Wrapper Induction for End-User Semantic Content Development. Proceedings of First International Workshop on Interaction Design and the Semantic Web. ISWC 2004.

- 4) Hutchings, D. and Stasko, J. QuickSpace: New Operations for the Desktop Metaphor. Extended Abstracts of the Conference on Human Factors in Computing Systems 2002.
- 5) North, C. and Shneiderman, B. Snap-together visualization: A User Interface for Coordinating Visualizations via Relational Schemata. Proceedings of the Working Conference on Advanced Visual Interfaces, 2000, p.128-135.
- 6) Prasad A.R.D. and Patel Dimple (2003). An Overview of Semantic Web.
- 7) Quan, D., Huynh, D., Karger, D. and Miller, R. User Interface Continuations. Proceedings of User Interface Software and Technology (UIST) 2003.
- 8) Rutledge, L., Houben, G. and Frasincar, F. Combining Generality and Specificity in Generating Hypermedia Interfaces for Semantically Annotated Repositories. Proceedings of First International Workshop on Interaction Design and the Semantic Web, ISWC 2004.

ONTOLOGY FOR SEMANTIC MULTIMEDIA WEB

Hiranmay Ghosh¹ and Santanu Chaudhury²

¹TCS Innovation Labs Delhi, TCS Towers, 249 D&E Udyog Vihar Ph-IV, Gurgaon 122015, INDIA

²Electrical Engineering Department, Indian Institute of Technology, Delhi, New Delhi 110016, INDIA

Abstract— This paper proposes a new multimedia ontology based scheme for semantic multimedia data processing on the web. The ontology language “Multimedia Web Ontology Language” (MOWL), is designed as an extension of OWL, the W3C recommended ontology language for the web. MOWL supports creation of and reasoning with perceptual modeling of concepts, and probabilistic evidential reasoning.

Index Terms— Multimedia systems, Ontology, Semantic Web, Bayesian network, Evidential reasoning

1. INTRODUCTION

The vision of semantic web proposes an environment where the data and services on the web can be semantically interpreted and processed by machines to facilitate human consumption. In today’s cyberspace, audio-visual artifacts compete with traditional text and data in their information content. Machine interpretation of multimedia data is therefore essential for realization of the semantic web vision. While textual documents are effectively processed by semantic web technology, application of the technology for multimedia data is still at its infancy. In this context, we propose a new ontology based approach for contextual semantic interpretation of multimedia data.

Semantic web technology relies on ontology as a tool for modeling an abstract view of the real world and contextual semantic analysis of documents. Ontology languages like Web Ontology Language (OWL)[1] uses linguistic constructs for modeling the real-world and can be conveniently used for interpreting textual documents. An attempt to use ontology for interpreting multimedia contents is hindered by the *semantic gap* that exists between media features appearing in the documents and the linguistic structures representing the concepts in the ontology. To cope up with this deficiency, there have been some attempts to extend ontology with addition of media examples for multimedia data processing [2]. However, such extensions do not meet the specific demands for reasoning with media data. Semantic retrieval in multimedia repository is generally carried out by intelligent concept recognition tasks that are specific to the media types and repository architectures. Multimedia libraries, pertaining to a common theme, can be built by several independent organizations in

different ways and call for different retrieval strategies. Integration of such heterogeneous collection under a common thematic umbrella is a challenge with current semantic web technology.

In this context, we propose to extend Multimedia Web Ontology Language (MOWL) [3] as an extension to OWL, with additional capability of creating media-based perceptual models of real-life concepts and events and to reason with them. In contrast to crisp Description Logics (DL) based reasoning in OWL, we propose probabilistic evidential reasoning to cope up with the uncertainties that are inherent to multimedia data processing. Further, it is possible to create unique repository specific search strategies in response to a semantic query by reasoning with the perceptual model and the repository capabilities.

The rest of the paper is organized as follows. Section 2 provides an overview of the perceptual reasoning model and justifies proposal of MOWL. The new language constructs for MOWL are introduced in section 3. Section 4 explains the evidential reasoning scheme with MOWL. Finally, section 5 concludes the paper.

2. PERCEPTUAL REASONING AND MOWL

Ontology is a formal description of the abstraction of a domain. Human beings use natural languages to communicate an abstract view of the world. Natural language constructs are symbolic representations of human experience and is close to the conceptual model that Semantic Web technologies deal with. Thus, it seems quite *natural* to use natural language constructs to represent the ontology elements. As a result, it becomes convenient to apply semantic web technologies in the domain of textual information. In contrast, media artifacts are perceptual recording of human experience. An attempt to use the conceptual model to interpret these perceptual records gets severely impaired by the semantic gap that exists between the perceptual media features and the conceptual world. However, the concepts have their roots in perceptual experience of human beings and the apparent disconnect between the conceptual and the perceptual worlds is rather artificial. The key to semantic processing of media data lays in harmonizing the seemingly isolated conceptual and the perceptual worlds.

Concepts are formed in human minds through a complex renement process of personal experiences. Observations of the real world objects amounts to reception of large volumes of perceptual data through our sensory organs. The raw data go through a process of renement to result in mental models. The models are further abstracted over a large number of observations, to give rise to *concepts*, which are labeled with linguistic constructs to facilitate communication. The fact that concepts are abstractions of perceptual observations has an interesting consequence. A concept gives rise to the expectation of some perceptible media properties on its embodiment in a multimedia artifact. Observation of those media properties forms the basis of concept recognition in a multimedia document. Figure 1 depicts the formation of the concept *Medieval Indian Monument* and the abstracted visual patterns that are expected on an embodiment of the concept in a multimedia artifact. Note that the different instances of the concept have significant variations and the perceptual model comprises an abstraction of their common visual properties.

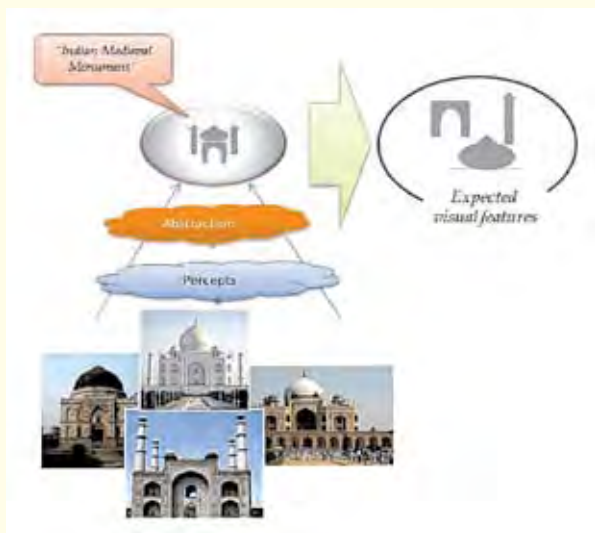


Fig. 1. Perceptual Model

Multimedia Web Ontology Language (MOWL) has been proposed to encode the domain knowledge pertaining to the expected perceptual properties of concepts and to reason with them, over and above the semantic properties that are encoded and reasoned with in OWL. Figure 2 depicts a small section of an ontology encoding media based description of concepts. The individual media patterns are often connected by some spatial or temporal relationships with each other, in context of a concept or an event. For example the dome of a medieval monument should occur above its other components, such as the minarets or the facade. moreover, the expected perceptual properties of concepts can be *inherited* by other concepts in the domain,

depending on their relationship. For example, a specic instance of a monument, the **Taj Mahal**, is *made of* a specic class of stones, **marble**, and therefore expected to “inherit” its color and texture properties. This form of media property inheritance is quite distinct from classical *property inheritance* rules that is supported by classical ontology models and requires distinct form of reasoning. This is required for creating a complete Observation Model of a concept using media properties of that concept and other related concepts.

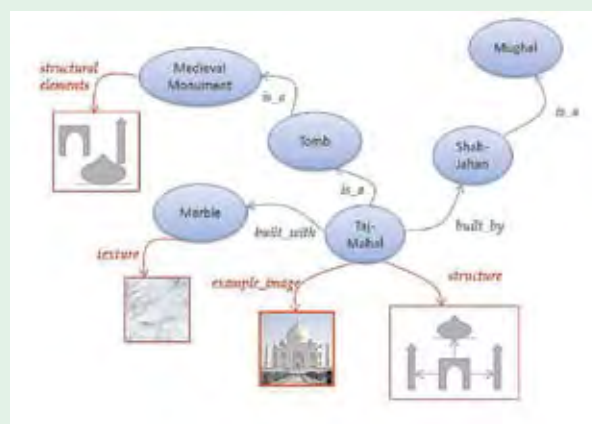


Fig. 2. Multimedia Ontology

Once an *Observation Model* for a concept is created, the presence of the expected media patterns and their spatiotemporal relationships can be veried in a multimedia artifact to detect the presence of the concept. However, there can be significant intrinsic differences between the instances of the concepts as well as the instances of the media artifacts depicting a specic instance of the concept. The latter can be attributed to variation in viewpoints, change in perspectives, occlusions and lighting conditions. This leads to a requirement of uncertain reasoning for semantic multimedia data processing. We have proposed a Bayesian Network based evidential reasoning scheme with MOWL.

3. MOWL LANGUAGE CONSTRUCTS

MOWL is designed as an extension of OWL to ensure compatibility with the W3C standards. It uses OWL constructs to dene classes, individuals and properties. In addition, it proposes some language extensions for encoding media properties, property propagation rules and speciation of conditional probabilities that characterizes uncertainty in association between concepts and their media properties.

The abstract class *mowl:MediaObject* in MOWL signiûes media properties. In general, the expected media properties of concepts can either be specied as

property constraints, e.g. *color = WhiteColor*, or be cited as media examples. Accordingly, MOWL defines two subclasses of *mowl:MediaObject* (see figure 3(a)):

1. Media property class *mowl:MediaFeature* can define a constraint on a media feature. For example *color* is *WhiteColor*.
2. Media example class *mowl:MediaExample* can define a media example. For example, photograph of a monument.

We propose that the media feature specifications follow MPEG-7 [4] schema.

The OWL class *owl:ObjectProperty* can be instantiated to give individual properties with a defined range and a domain. Media features are associated with concepts. Hence the domain and the range of an instantiated property are set as a *mowl:MediaFeature* and the concept respectively. For example, we associate the media feature *color=WhiteColor* with a concept **marble** using an instantiated property *hasMediaFeature*. Similarly, we can associate an example image with the concept **TajMahal** using an instantiated property *hasMediaExample* (see figure 3(b)).

Media properties and examples can propagate across connected concepts in an ontology depending on the semantics of the relation in a specific domain. MOWL defines the properties *mowl:FeaturePropagationProperty* and *mowl:ExamplePropagationProperty* to indicate the two types of propagation. A media feature or an example will propagate from one concept to another if and only if the relation connecting the concepts has the corresponding property (See figure 3(c)). For example, if the relation *builtWith* has *mowl:FeaturePropagationProperty* and connects the concepts **TajMahal** and **marble**, the properties of marble will flow into TajMahal.

The different media properties may be connected with different spatio-temporal relations. MOWL defines constructs for specifying such spatio-temporal properties (see figure 3(d)). Rather than restricting the relations to a few pre-specified values, such as *left*, *right*, *before*, *after*, etc., MOWL provides for defining such relations. One way to define such relations has been proposed in [5].

The probabilistic reasoning model of MOWL is based on Bayesian Network. Accordingly, MOWL

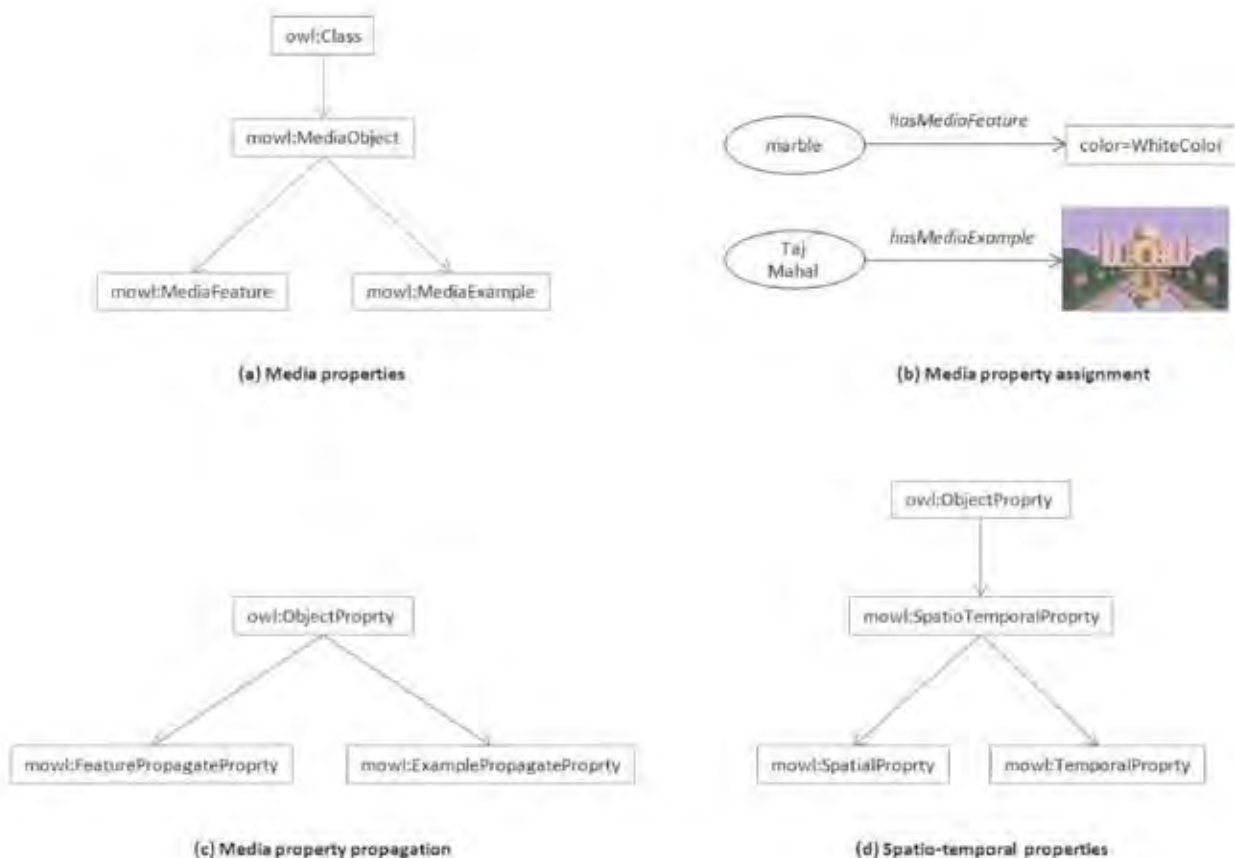


Fig. 3. MOWL Constructs

treats the nodes (concepts) in the ontology as random variables and defines constructs for specifying Conditional Probability Tables (CPT) for the connected nodes. The approach is similar to [6] that proposes a probabilistic extension of OWL.

4. REASONING WITH MOWL

In our approach, concept recognition in multimedia artifacts is based on observation of some expected media patterns in the artifact. We model concept recognition as an evidential reasoning problem. If a concept c causes a set of media patterns M to appear in a multimedia artifact, observation of a specific pattern $m \in M$ provides some evidence towards the concept. A concept is recognized when there is sufficient cumulative evidence for the concept as a result of observation of several media features, $m_1, m_2, \dots, m_k \in M$. There are three distinct stages of reasoning with MOWL for semantic query processing.

The first stage involves derivation of an Observation Model for a semantic query. A semantic query is mapped to one or more MOWL concept nodes based on some similarity measure between the query and the attributes of the node attributes. An Observation Model for each of these nodes is created from the media properties associated with that node and other nodes which are connected to it with a relation bearing *propagate* property. The CPT's for the connected nodes are derived from the corresponding nodes in the ontology. Finally, the Observation Models for all the nodes are merged. The resultant Observation Model is thus organized as a Bayesian Tree. The root node in the tree represents the concept and the leaf nodes represent its expected media properties. Figure 4 depicts a typical Observation Model for the concept **TajMahal**.

The Observation Model is neutral to any specific media type or repository architecture and, in general, contains a redundant set of patterns for different types. It may not be possible or may be extremely computationally expensive to evaluate all of these media patterns at a given multimedia repository. For example, an audio pattern cannot be detected in a library of still images. However, the evidential reasoning scheme deployed with MOWL can produce robust results despite limited input data. This property is exploited to realize concept recognition with a subset of media patterns that is specified in an Observation Model. Identification of an optimal subset of the media patterns to formulate a search strategy for a multimedia artifact collection is described in [7].

A retrieval strategy which is a subgraph of an Observation Model, is also organized as a Bayesian

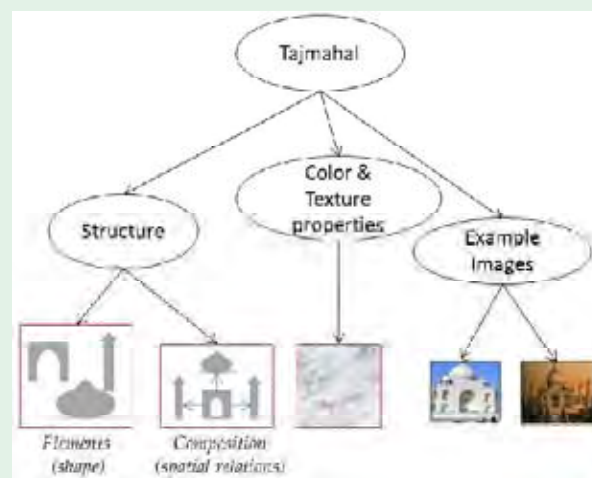


Fig. 4. Observation Model

Tree with the concept at the root and the expected media patterns in the leaf nodes. An *observation* with a leaf node in a retrieval strategy involves actuation of some feature detection mechanism to discover the corresponding media feature in a multimedia artifact. In general, an observation results in a similarity score, which is normalized in the range $[0, 1]$. We interpret this similarity score as a virtual evidence for the leaf node. The assignment of the virtual evidences to the different leaf nodes result in belief propagation in the Bayesian Network. The posterior probability of the root node as a result of such belief propagation represents the degree of belief in the concept.

5. CONCLUSION

In this paper, we have proposed a new ontology representation suitable for processing multimedia data repositories distributed on the web. The ontology representation MOWL is a syntactic extension of OWL, the current standard of ontology language for the web proposed by W3C. It supports perceptual modeling of concepts and a probabilistic evidential reasoning scheme that is necessary for multimedia data processing.

6. REFERENCES

- [1] W3C Recommendation, "OWL Web Ontology Language: Overview," February 2004.
- [2] Marco Bertini, Alberto Del Bimbo, Carlo Torniai, Costantino Grana, and Rita Cucchiara, "Dynamic pictorial ontologies for video digital libraries annotation," in *MS '07: Workshop on multimedia information retrieval on The many faces of multimedia semantics*, New York, NY, USA, 2007, pp. 47–56, ACM.
- [3] Hiranmay Ghosh, Santanu Chaudhury, Karthik Kashyap, and Bridaduti Maiti, *Ontology Specification and Integration for Multimedia Applications. In Ontologies:*

A Handbook of Principles, Concepts and Applications in Information Systems, pp. 265–296, Springer, 2007.

- [4] B S Manjunath, Phillipe Salembier, and Thomas Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, 2002.
- [5] Sujal Subhash Wattamwar and Hiranmay Ghosh, “Spatio-temporal query for multimedia databases,” in *MS '08: Proceeding of the 2nd ACM workshop on Multimedia semantics*, New York, NY, USA, 2008, pp. 48–55, ACM.
- [6] Zhongli Ding and Yun Peng, “A probabilistic extension to ontology language OWL,” in *Proceedings of the 37th Hawaii International Conference On System Sciences*, January 2004.
- [7] Hiranmay Ghosh and Santanu Chaudhury, “Distributed and reactive query planning in r-magic: An agent based multimedia retrieval system,” *IEEE Trans KDE*, vol. 16, September 2004.

LISTING VARIOUS TOOLS IN SEMANTIC WEB: A SUMMARY

Jaya gupta¹, Sanjay Kumar Malik²

1M.Tech-IT, 2nd Sem, University School of Information Technology, GGS Indraprastha University, Delhi

2Asstt. Professor, University School of Information Technology, GGS Indraprastha University, New Delhi

1jaya_gupta8@yahoo.com, 2sdmalik@hotmail.com

Abstract— The concept of Semantic Web is originated by Sir Tim Berners-Lee by expressing the basic vision as "A meaningful Web in which computers become capable of analyzing the data on the Web, by providing the content, links, and transactions between people and computers". The Semantic Web gives the idea of having data which is defined and linked on the Web in the way that it can be used by machines not just for display purpose but also for development, integration, and reuse of data across various applications. To accomplish this goal, the various tools are being developed for assisting in Semantic Web research projects. These tools may be effective and helpful to build, manipulate, interrogate or enrich the Semantic Web. There are large number of tools available like developers can use ontology editing tools e.g. protégé to create, manipulate, import & export of ontologies.

Researcher or users need to know about the various existing tools being used in Semantic Web which may support effective sharing of information, distributed knowledge to integrate and re-used in Semantic Web which helps in knowledge management. This paper summarizes various tools which are being used in the Semantic Web and it may help researchers to choose a tool required for their application or need towards semantic web.

Keywords: Semantic Web, ontology, Semantic Web tools.

1. INTRODUCTION

The Semantic Web is the Web of data whose fundamental principle is the creation and use of semantic metadata. Various tools have been developed and are being developing in the ongoing semantic web research projects. These tools may help in overall semantic web development or ontology development which supports various applications and help in knowledge management. These tools often provide easy to use functionality, environment for consistence checking, promote easy and fast navigation between concepts, have tutorial support, and offers Plug-ins [4].

2. CATEGORIZATION OF VARIOUS TOOLS

We can categorize these tools according to their functionality as shown in figure 1 and its other features are shown in table below.

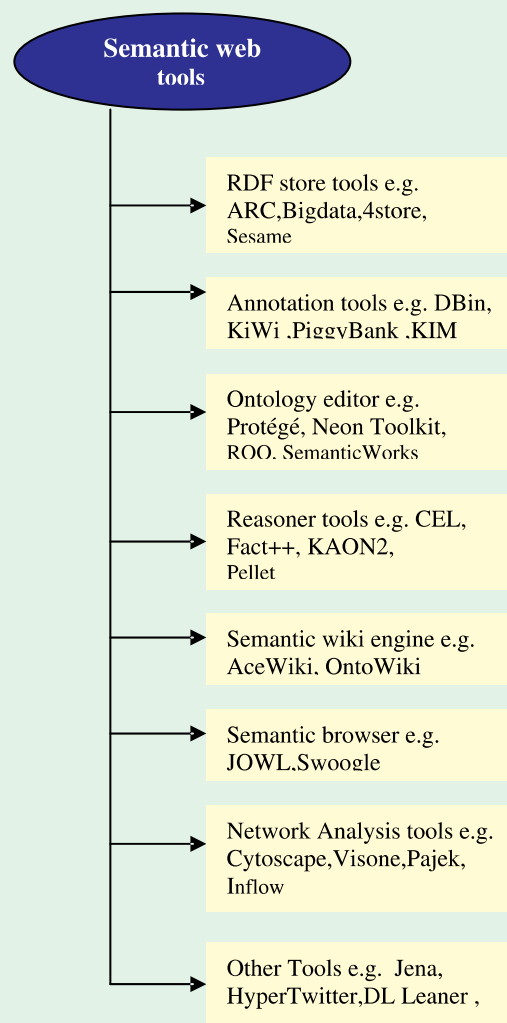


Figure 2: Semantic Web Tools

Table : A Comparative study of various Semantic Web tools

RDF STORE TOOLS - It stores and manages any kind of RDF data

Name of tool	Version	Purpose	Features	Based on	Released	Website
ARC	2	RDF Store & semantic web development	Flexible RDF storage, SPARQL query	PHP	05.03.09	http://arc.semsol.org/
Bigdata	0.81b	RDF store	Supports RDFS and OWL Lite reasoning , high-level query	Java	26.10.09	http://www.systap.com/bigdata.htm
Sesame	3.0-alpha1	RDF store, Querying	Open source RDF framework, support RDF Schema inferencing and querying	Java	23.02.09	http://www.openrdf.org/
4 store	0.9.2	RDF store & query	Efficient, scalable & stable RDF database	ANSI C99	14.07.09	http://4store.org/

ANNOTATION TOOLS - It supports semantic annotation of all kinds of content

Name of tool	Version	Purpose	Features	Based on	Released	Website
DBin	Alpha 2.0	Semantic annotation tool ,content management	Creates discussion group where users annotate any subject of interest	Java	10.01.08	http://www.dbin.org/
KiWi	0.5	Semantic annotation tool, Semantic wiki engine	Knowledge in a wiki combines the wiki philosophy	JBoss Seam, Java EE	01.07.09	http://www.kiwi-project.eu/
PiggyBank	3.1	Semantic annotation tool, Semantic browser	Firefox extension that turns your browser into a mashup platform	Java	04.09.07	http://simile.mit.edu/wiki/Piggy_Bank
KIM	2.3	Semantic annotation, Indexing, Retrieval	Provides services for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured content	Java	2008	http://www.ontotext.com/kim/KIM-downloads.html

ONTOLOGY EDITING AND DEVELOPMENT TOOLS - It builds, create, manipulate and edit ontologies, helps in ontology documentation, ontology export and import

Name of tool	Version	Purpose	Features	Based on	Released	Website
Internet business logic	1.0	Ontology editor	Kind of wiki and SOA Endpoint for writing and running business rules in open vocabulary	Java	18.05.09	http://www.reengineeringllc.com/
NeON toolkit	v2.3	Ontology editor	Performs ontology engineering activities	Java/ Eclipse	08.02.10	http://neon-toolkit.org/wiki/Main_Page
ROO	1.0.1	Visual RDF & OWL editor	OWL ontology construction tool	Java	13.05.09	http://www.comp.leeds.ac.uk/confluence/
Protégé	3.4.3	Ontology editor	Free, open source ontology editor and knowledge-base framework.	Java	03.02.10	http://protege.stanford.edu

Semantic Works 2010	v2010r2	Semantic Markup, Ontology, and RDF Editor	Graphical RDF,RDFS,OWL editor	Java	17.11.09	http://www.altova.com/semanticworks.html
Top Braid composer	2.6.2	Ontology editor	Powerful graphical development environment for modeling data	Java/ Eclipse	04.08.08	http://www.topquadrant.com/products/TB_Composer.html
OWL API	3.0.0	Ontology Editor	Create, manipulate & serialising OWL Ontologies	Java	28.01.10	http://owlapi.sourceforge.net/

REASONER TOOLS – It helps to infer logical consequences from a set of asserted facts

Name of tool	Version	Purpose	Features	Based on	Released	Website
CEL	1.0 Build 8	Reasoner	Reasoner for the polynomial description logic	Java	05.02.09	http://lat.inf.tu-dresden.de/systems/cel/
Fact++	1.2.3	Reasoner tool	Well-known FaCT OWL-DL reasoner	C++	05.03.09	http://owl.man.ac.uk/factplusplus/
HermiT	1.2.1	Reasoner	First publicly-available OWL reasoner	Java	12.12.08	http://www.hermit-reasoner.com/
Pellet	2.0 RC6	Reasoner	Standard & cutting-edge reasoning services for OWL	Java	30.04.09	http://clarkparsia.com/pellet

SEMANTIC WIKI ENGINE TOOLS – Provides ability to capture information about data within pages

Name of tool	Version	Purpose	Features	Based on	Released	Website
AceWiki	0.3.1	Semantic wiki engine	Powerful semantic wiki using ACE, predictive editor	Java	21.01.10	http://attempto.ifi.uzh.ch/acewiki/
SMW+	1.4.6	Semantic annotation tool, Semantic wiki engine	Semantic enterprise wiki that lets you tag, process and query data	Java	21.12.09	http://wiki.ontoprise.com/smwforum/index.php/Main_Page
OntoWiki	0.9	Semantic wiki engine	Tool providing support for agile, distributed knowledge engineering	PHP	09.04.10	http://ontowiki.net/Projects/OntoWiki

SEMANTIC SEARCH ENGINE TOOLS – It improves search accuracy, search queries in exact form with relevant results

Name of Tool	Version	Purpose	Features	Based on	Released	Website
JOWL	1.0	Semantic browser, Semantic search tool	Allows client side (live) loading and visualization of OWL DL	Java	March 2009	http://jowl.ontologyonline.org/
Swoogle	Swoogle 2006	Search engine	Search engine for Semantic Web documents, terms and data found on the Web.	Java	2006	http://swoogle.umbc.edu/

NETWORK ANALYSIS TOOLS – Used to identify, represent, analyze, visualize and simulate nodes and edges

Name of tool	Version	Purpose	Features	Based on	Released	Website
Cytoscape	2.6.3	Network visualizing tool	Open source ontology visualization & analysis tool	Java	01.07.09	http://www.cytoscape.org/
Pajek	1.26	Network Analysis and Visualization	Analysis and visualization of large networks	Delphi (Pascal)	01.01.10	http://pajek.imfm.si/doku.php
Inflow	3.1	Network Analysis and Visualization	Excels at what-if analysis, change the network, get new metrics	Java	-	http://www.orgnet.com/inflow3.html
Visone	2.5.1	Network Analysis and Visualization	Interactive analysis and visualization of social networks	Java	26.08.09	http://visone.info/

OTHER TOOLS – Helps in building web applications, framework, Microblogging, Machine learning in OWL

Name of tool	Version	Purpose	Features	Based on	Released	Website
CubicWeb	3.6.11	Semantic Web development	Build web applications by reusing components called cubes	Python	26.02.10	http://www.cubicweb.org/
Jena	2.6.0	Semantic Web development, RDF store	Framework for building Semantic Web applications	Java	16.10.09	http://jena.sourceforge.net/
DL-Learner	Build 2009-05-06	Semantic Web tool, Machine Learning tool	Tool for supervised Machine Learning in OWL and Description Logics.	Java	06.05.09	http://dl-learner.org/Projects/DLLearner
HyperTwitter	0.9	Semantic MicroBlogging	Embedding triple-like statements into Twitter microblogging messages	Python	23.02.10	http://http://semantictwitter.appspot.com/

3. CONCLUSION AND FUTURE WORK

The Semantic Web provides opportunities for users to get better search results and enables people to share content on web and comprises of set of principles, various tools, techniques and collaborative work which will fulfill the need of the web. This paper may help researchers or users to summarize various tools and to choose an appropriate tool according to their applications or requirements. The future work may be using some of the mentioned tools towards the realization of Semantic Web.

4. REFERENCES

- [1] Obitko:tutorials.ontologies-semantic-web. Available at <http://www.obitko.com/tutorials/ontologies-semantic-web/ontologies.html>
- [2] John Hebel, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez. Semantic Web Programming.
- [3] Berners-Lee, Tim (May 1, 2001). "The Semantic Web". Scientific American.
- [4] Jorge Cardoso. University of Madeira, Portugal. Semantic Web Services: Theory, Tools, and Applications. Page numbers 71-74

- [5] W3C. W3C Semantic Web Activity. Available at <http://www.w3.org/TR/owl-features/>
- [6] Protégé. Welcome to protégé. Available at <http://protege.stanford.edu/>
- [7] DARPA (2006) DAML: The DARPA Agent Markup Language. Semantic web tools tutorials. available at <http://www.daml.org/2003/05/swmu-tools-tutorial/Overview.html/>
- [8] Semanticweb.org (2009) The Semantic Web. Available at <http://semanticweb.org/wiki/tools/>
- [9] D Shivalingaiah, Umesha Naik. 7th International CALIBER 2009. INFLIBNET centre. Semantic Web Tools: An Overview.
- [10] Jennifer Golbeck, Michael Grove, Bijan Parsia, Aditya Kalyanpur, and James Hendler. Maryland Information and Network Dynamics Laboratory University of Maryland, College Park College Park, Maryland, 20742, USA. New Tools for the Semantic Web.
- [11] AI³. Adaptive Information, Adaptive Innovation, Adaptive Infrastructure. Available at <http://www.mkbergman.com/category/semantic-web-tools/>

ANALYZING SEMANTIC WEB BY PRINCIPLE OF RATIONALITY (SOAR-BASED COGNITIVE ARCHITECTURE)

Neeranjan Chitare¹, Shrestha Rath¹

¹International School of Information Management, University of Mysore, Mysore
neeranjan@isim.net.in, shrestha_rath@isim.net.in

Abstract— The Semantic Web is an idea that the Web as a whole can be made more intelligent and perhaps even intuitive about how to serve a user's needs. In other words an effort is made to make the Web understand different notions of human cognition thereby providing a deep insight to synchronise human way of thinking and computers' way of analysing data. Hence human cognition plays a significant role in evolving semantic web with respect to the ever growing data.

Cognitive architecture is based on the theory of human cognition having a large selection of human experimental data, and is implemented as a running computer simulation program. It is of great importance in Human Computer Interface (HCI) which includes attention, problem solving, decision making, and so on. SOAR (State, Operator and Result) is a symbolic cognitive architecture. It is used to model different aspects of human behavior. It describes Perception or Motor Interface (PMI) from the external to internal representation in working memory and vice-versa. SOAR keeps semantic knowledge in the long term memory of its architecture. When choosing different paths in the problem space, knowledge comes in to place and is known as Principle of Rationality. Principle of Rationality interpenetrates every aspect of the SOAR architecture from operator selection to reinforcement learning.

In this paper we are trying to analyze web and semantics using SOAR. The development of semantic web not only comprise of human perceptions towards it but also human reflections for the existing semantic web. In addition to this, the action of agent is governed by Principle of Rationality. Here, the pre-existing memory which can be long term or short term memory manipulates the action of agent. We can study different actions of agent with the help of SOAR. This operation of agent may or may not be user-defined depending on the prevailing condition. As a consequence of which semantic web is affected by Principle of Rationality in different aspects like optimizing search, path selection and so on.

Keywords-- cognitive architecture, SOAR, HCI, PMI, principle of rationality

I. INTRODUCTION

In today's world, semantic web plays a crucial role. Specially, when it comes to application like searching operations different perceptions of semantic are vitally important. Through semantic web, we focus on involving advanced features in web. These features promote the existing web towards more interactive and more intelligent to solve user's needs. It anticipates a number of ways in which single or multiple groups can use self-explanations and other techniques, so that situation-handling programs can purposefully find what the users want. [1]

A cognitive architecture endeavors to combine different artificial intelligence and psychological theories together in a very standardized way in order to create intelligent systems to cope with the contemporary real world problems. However, a cognitive architecture is a theory on how human cognition works. Cognitive architectures differ from each other based on varied behavior of the human being, which is determined through psychological experiments. [2] So, in all, cognitive architecture is defined as the architecture which is based on the theory of human cognition having a huge selection of experimental data related to human, and is implemented as a running simulated computer program. In Human Computer Interface (HCI), attention, problem solving, decision making are of prime importance.

II. COGNITIVE ARCHITECTURE FOR SEMANTIC WEB

Soar, which is a type of cognitive architecture, is generally known as a search through a problem space in which an operator can be applied to a state to get a result. [2] Fig. 1 illustrates semantic model of Soar architecture. Following the above Fig1 we can state that as different memories are part of semantic model, in the same manner semantic memory also plays a crucial role in application of semantics. [3] The model is generally goal-oriented, so its implementations focus on goals and sub goals to complete successfully the desired tasks. The tasks include various complexities such as choosing methods of solving, ranking and relevance when it comes to searching operations, etc. To

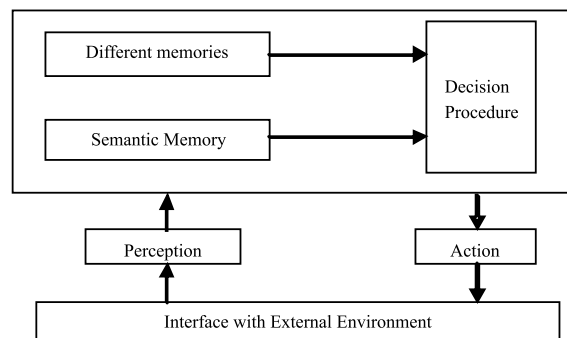


Fig. 1 Semantic model

complete the appropriate tasks, description of complex environments, processing of large amounts of knowledge and generalization of objects are necessary. Flexibility and ability of learning helps to run the problems and to deal with them. [2]

Problem spaces are one of the important issues to achieve the goals in Soar. Every goal is comprised of a number of sub-goals which together find out a number of ways to complete a particular goal. A variety of paths are needed to solve for a specific goal.

III.ROLE OF PRINCIPLE OF RATIONALITY IN SEMANTIC WEB

In general, for any situation, there is a defined problem space and defined number of possible solutions. When choosing distinct paths in the problem space, knowledge comes into place. This is called the 'principle of rationality'. [2] Thus in the state of impasse, the pre-existing knowledge helps to choose proper and efficient path leading to solution.

In Semantic web, imposing semantics and extracting semantics both play a crucial role and are equally important for development of semantic web. With the advent of time and meticulous observations in semantic web world, various possible options have emerged. These options are intended to provide an assist, the way we approach to reach the goal while performing a particular task. In addition to this, the options simplify and optimize the way we approach for a particular task. Soar is having semantic knowledge to execute the present goal in the working memory. This knowledge is stored in the architecture's long-term memory.

When we consider Soar as the cognition model for semantic web, it describes the PMI (Perception/Motor Interface) for "defining mappings from the external world to the internal representation in working memory, and from the internal representation back out to action in

the external world". Thus, PMI is the interface between working memory and the external world. [2]

During operation, the decision cycle is split up into two parts: elaboration and decision. The elaboration part activates all associations which connect the working memory with long-term memory. Stronger associations are charged a number of times, and then opted chunks are placed in separate slots of the working memory, and then valued the actual decision cycle. Stronger the association between a slot and an operator, the more likely the action takes place. In the end, the working memory of the single slot is altered with new operator, a new state, a new problem space or either a new goal. A conventional overview of this process can be seen in fig2.

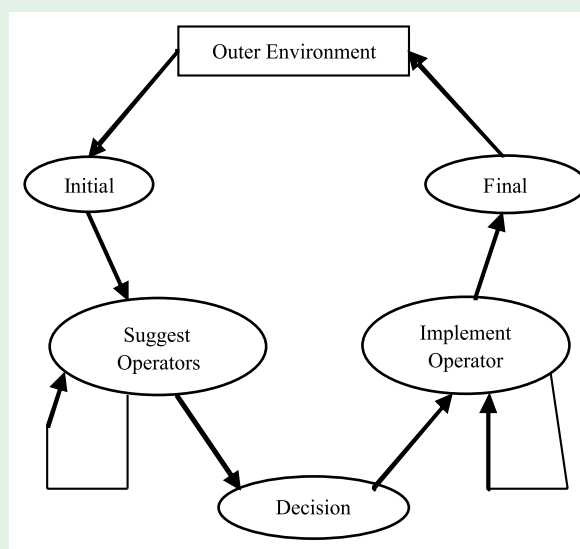


Fig. 2. Soar decision making model

IV. IMPOSING HCI PATTERN OF "CHUNKING" IN SEMANTIC WEB

Learning generalizes combining what happens now and what happened in the past. These combinations are called chunks, and the process of creating chunks is called chunking. Chunking occurs when impasses are solved deriving a solution for considering an option, this option is then stored in Long-Term Memory. In Soar, chunking is the basis of all other types of learning. Besides learning, Soar is also able to forget chunks by using destructive operations.

It is Soar's learning mechanism that converts the results of problem solving in sub-goals into rules. It compiles knowledge and behavior from deliberate to reactive. Although chunking is a simple mechanism, it is extremely general and can learn all the types of knowledge encoded in rules. [4]

In accordance with this, in semantic web when we go for imposing or extracting of semantics establishing relationship is must.

For example, in Google search engine [5], if we have to search for keyword 'water', then just type 'water' in the search text box. Here we can see that as soon as we type 'water', ten suggestions will come in the drop-down menu of the search text box. The suggestions will be as follows – 'water pollution', 'water purifier', 'waterfall model', 'water conservation', 'water cycle', 'water resources', 'water kingdom' and so on. These suggestions are the most frequently used for that particular keyword in that particular time period. These suggestions will go on changing from time to time and from day to day depending on the requirements. For phrase search, let us take an example of 'water in coco cola', as soon as we write this in search text box, 'percentage of water in coca cola' will appear in drop-down menu search text box. Thus we come to know the role of semantic memory for successful implementation in semantic web environment.

V. CONCLUSION

Soar as a cognitive architecture plays a decisive role for imposing or extracting semantics, thereby applying its Principle of Rationality in semantic web. By analyzing the cognition which is implemented by semantic web, we can state that for advancing semantics it is must to precisely come with possible options in a defined problem space. Proper and systematic organization of semantic memory in a module can enhance the performance in different applications.

REFERENCES

- [1] SearchSoftwareQuality. [Online]. Available:http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci214349,00.html
- [2] WA de Landgraaf, Lehman (1993) "Implementing the Mind-Cognitive architectures"
- [3] John E. Laird "Extending the Soar Cognitive Architecture"
- [4] Steier, D.S., Laird, J.E., Newell, A., Rosenbloom, P.S., Flynn, R., Golding, A., Polk, T.A., Shivers, O., Unruh, A., Yost, G.R. (1987) Varieties of Learning in Soar. Proceedings of the Fourth International Machine Learning Workshop.
- [5] The Google website. [Online]. Available:<http://www.google.co.in/>

Semantic Annotation Tools for Knowledge Management: Analysis and Review

***Pooja Kherwa**

Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi - 110058

Sanjay Kumar Malik

*Assistant Professor, GGSIP University, Kashmere Gate,
New Delhi - 110006*

Abstract– Support for information and knowledge exchange is a key issue in the information society. To reduce the time wasted in searching and to reduce associated user frustration much more selective user access is needed. This is possible by semantic information processing of online documents.

Knowledge management in an organisation are used for managing knowledge resources in order to facilitate access and reuse of knowledge.

Semantic annotation is about assigning to the entities in the text, links to their semantic descriptions. This sort of metadata provides both class and instance information about the entities. Semantic annotation is applicable for any type of text-web pages, regular documents etc.

For semantic annotation, there are various manual, semiautomatic and full automatic tools are developed by various organizations like mindswap.org, ontotext.org etc.

In this paper, we are presenting analysis and review of some of these tools according to their applicability for an application domain in knowledge management.

Our review and analysis may help the research community in selecting an appropriate tool for extracting the relevant and desired information from huge knowledge base of an organisation.

I. INTRODUCTION

The semantic web purposes annotating document content using semantic information. The result is web pages with machine interpretable markup to create annotation with well defined semantics.[1]

Manual Annotation

Manual annotation is more easily accomplished today, using authoring tools such as Semantic Word[2], which provide an integrated environment for simultaneously authoring and annotating text. However, the use of manual annotation is often leads with errors due to factors such as annotator familiarity with the

domain, amount of training, personal motivation and complex schemas [3]. Manual annotation is also an expensive process

Another problem with manual annotation is the volume of existing documents on the Web that must be annotated to become a useful part of the Semantic Web.

Semiautomatic Annotation

Semiautomatic annotation of documents has been proposed. Semiautomatic means, as opposed to completely automatic, are required because it is not yet possible to automatically identify and classify all entities in source documents with complete accuracy[4]. All existing semantic annotation systems rely on human intervention at some point in the annotation process [5].

Automated annotation

Automated annotation provides the scalability needed to annotate existing documents on the Web, and reduces the burden of annotating new documents. Other potential benefits are consistently applying ontologies, and using multiple ontologies to annotate a single document.

As a motivating example of what can be achieved once documents are given semantic markup. Consider the medical imaging and advanced knowledge technologies (MIAKT) project. MIAKT has developed problem environment for use in the medical domain. In MIAKT the annotation make the knowledge contained in unstructured sources (x-ray available in structured form, allowing both accurate and focused retrieval and knowledge sharing for a given patients) Moreover the annotation can be used to provide automated services, for e.g., they can be processed using natural language generation software to automatically draft textual report about the patient, the diagnostic information that is available and assessment made about the data by the medical team, a task which usually consumes doctors' valuable time[6].

The paper is organized as follows. In section 2, we are discussing about analysis criteria for semantic

annotation tools, In section 3, we are presenting our analysis and review in tabular form according to the criteria discussed in section 2, In section 4, we have conclude the paper by adding feature that should be supported by the future generation semantic annotation tools.

II. ANALYSIS CRITERIA FOR SEMANTIC ANNOTATION TOOLS

Analysis criteria that we have chosen for our review process of semantic annotation tools are overlap to some extent with the criteria set out by Handschuh et al.[7]. Analysis criteria for semantic annotation tools can be categorised as follows:

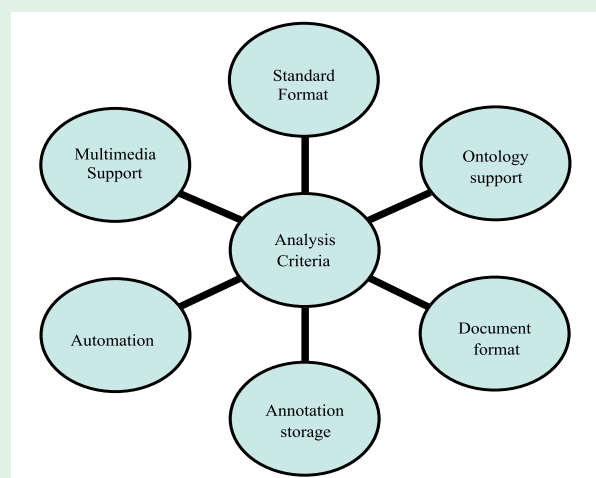


Fig. 1. Analysis criteria for semantic annotation tools

Standard format

For annotation systems in particular ,standards can provide abridging mechanism that allows heterogenous resources to be accessed simultaneously and collaborating users and organizations to share annotations.It is the activity of the w3c in developing and promoting international standards for the semantic web.

Ontology support

Second criteria is concerned with whether our annotation tool is supporting appropriate ontology format or not.ontology support is also concerned with the problem of ensuring consistency between ontologies and annotation with respect to ontology changes.

Support for heterogeneous document format

Semantic web standards for annotation tend to

assume that the document being annotated are in web native formats such as HTML and XML.In knowledge bases of organization we found documents in many different format including word processor files, spreadsheets, graphics files.So this is important for annotation tool, whether it support multiple document format or not.

Annotation storage

Another important criteria is where annotation will be stored,the semantic web model assumes that annotation will be stored separately from the original document ,whereas the “word processor model assumes that comments are stored as an integral part of the document.

Automation

Another important criteria is annotation of document or knowledge base is automatic or manual.

Multimedia support

Multimedia annotation is the next phase of development for annotation,expanding the range of files types like images,video,audio.that can be annotated with the annotation tools.

Analysis and Review of semantic annotation tools: Using analysis criteria as discussed above

CONCLUSION

In this paper,we have present an analysis and review of semantic annotation and its various tools ,which may help the researcher in selection of appropriate tool for their application domain.

Intelligent documents created by semantic annotation,would bring the advantages of semantic search and interoperability.our review of existing semantic annotation system indicates that although today we have lots of semantic annotation tools are available, each of them are fully enriched with annotation features. Still there is requirement for semantic annotation tools with more enriched features like:

Linguistic annotation,
 Commentry annotation,
 Natural language processing support,
 to create a more intelligent knowledge management system

Table 1:analysis and review[17]

Annotation tool	Standard format	Ontology support	Document format	Annotation storage	Automation	Multimedia Support
Amaya [8]	RDF(S) Xlink,Xpointer	Annotation Server	HTML, XHTMLand XML	Local or annotation server	No	No
Mangrove[9]	RDF	Annotation server	HTML,Email	RDF database	No	No
Vannotea[10]	XML	Annotation Server	MPEG-2, JPEG2000, Direct3D	Annotation server	No	Yes
OntoMat[6]	DAML,OIL, OWL, SQL	OntoBroker annotation inference server	HTML,DeepWeb	Annotation Server,embedded in webpage,separate file	Yes	No
M-Ontomat- Annotizer[11]	XML,RDF(S), DOLCE	OntoBroker annotation inference server	MPEG-7	Annotation server	Yes	Yes
SMORE	RDF(S)	Ontology server and ontology editing	HTML,text ,email and images	Embedded in webpage	Yes	Yes
Open ontology forge [12]	RDF(S),XML,Xlink, Xpointer	Local, editable ontologies	HTML, text, images	Local RDF or XML file	Yes	Yes
MnM[13]	RDF(S),DAML+OIL	Ontology server	HTML,Text	Embeded in webpages	Yes	No
PANKOW[14]	HTML	Pattern Based Annotation	HTML	RDF knowledge Base	Yes	No
KIM[15][16]	RDF(S), OWL	KIMO,	HTML	RDF knowledge Base	Yes	No

REFERENCES

- [1] T.Berner-Lee,J.Hendler,O.Lassila,The Semantic Web,Sci. Am.(2001)34-43.
- [2] Tallis, M., Semantic Word Processing for Content Authors in Second International Conference on Knowledge Capture, (Sanibel, Florida, 2003).
- [3] Bayerl, P.S., Lungen, H., Gut, U. and Paul, K.I., Methodology for reliable schema development and evaluation of manual annotations in Workshop on Knowledge Markup and Semantic Annotation at the Second International Conference on Knowledge Capture (KCAP), (2003).
- [4] Yesilada, Y., Harper, S., Goble, C. and Stevens, R.,Ontology Based Semantic Annotation for Enhancing Mobility Support for Visually Impaired Web Users in K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation, (2003).
- [5] Maedche, A. and Staab, S. Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 16 (2). 72-79.
- [6] K.Bontcheva,Y.Wilks,Automatic report generation from ontologies:The MIAKT approach,in:Proceeding of the 9th international Conference On Applications of Natural Language to information systems.
- [7] S. Handschuh, S. Staab, R. Studer, Leveraging metadata creation for the Semantic Web with CREAM, KI '2003—advances in artificial intelligence, in: Proceedings of the Annual German Conference on AI,September 2003, 2003.
- [8] V. Quint, I. Vatton, An Introduction to Amaya, W3C NOTE 20-February-1997, 1997 (<http://www.w3.org/TR/NOTE-amaya-970220.html> accessed on 28 July 2004
- [9] L. McDowell, O. Etzioni, S. Gribble, A. Halevy, H. Levy, W. Pentney,D. Verma, S.Vlasheva, Enticing ordinary people onto the SemanticWeb via instant gratification, in: Proceedings of the 2nd International Semantic Web Conference (ISWC 2003), October 2003,2003.
- [10] R. Schroeter, J. Hunter, D. Kosovic, Vannotea, A collaborative video indexing, annotation and discussion system for broadband networks, in: Proceedings of the K-CAP 2003 Workshop on "Knowledge Markup and Semantic Annotation", October 2003, Florida, 2003.

- [11] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouaras, Y. Avrithis, S. Handschuh, Y. Kompatsiaris, S. Staab, M.G. Strintzis, Semantic annotation of images and videos for multimedia analysis, in: Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), 29 May–1 June 2005, Heraklion, Greece, 2005.
- [12] N. Collier, A. Kawazoe, A.A. Kitamoto, T. Wattarujeekrit, T.Y. Mizuta, A. Mullen, Integrating deep and shallow semantic structures in open ontology forge, in: Proceedings of the Special Interest Group on Semantic Web and Ontology, vol. SIG-SWO-A402-05, 2004.
- [13] Vargas-Vera M., E. Motta, J. Domingue, M. Lanzoni, A. Stutt, F. Ciravegna, MnM: A tool for automatic support on semantic markup, KMi Technical Report, September 2003, TR Number 133, 2003.
- [14] P. Cimiano, S. Handschuh, S. Staab, Towards the self-annotating web, in: Proceedings of the 13th International World Wide Web Conference (WWW 2004), May 17–22, 2004, New York, NY, 2004.
- [15] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov, Towards Semantic Web information extraction, in: Proceedings of the Human Language Technologies Workshop at 2nd International Semantic Web Conference (ISWC2003), 20 October 2003, Florida, USA, 2003.
- [16] B. Popov, A. Kirayakov, D. Ognyanoff, D. Manov, A. Kirilov, KIM—a semantic platform for information extraction and retrieval, Nat. Lang. Eng. 10 (3/4) (2004) 375–392.
- [17] Victoria Uren a, Philipp Cimiano b, Jos´e Iria c, Siegfried Handschuh d, Maria Vargas-Vera a, Enrico Motta a, Fabio Ciravegna c, Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web (2005) .

SemArtha : An OWL based Ontology for efficient E-Governance

Sanghamitra Mohanty, Sohag Sundar Nanda, Soumya Mishra

PG Department of Computer Science and Application Utkal University, Vani Vihar, Bhubaneswar-751004, Odisha, India Sangham1@rediffmail.com Nanda.sohagsundar@gmail.com Soumyalitun@gmail.com

Abstract — In this paper we propose an ontology for E-Governance based on W3C recommended Web Ontology Language(OWL). An ontology is a form of knowledge representation that relates concepts within a domain. It helps in machine understandability of the meaning of data. Since the representation is semantic in nature, interoperability among stakeholders using different forms of the same data can be achieved. A semantic mediation device is used for this purpose. We discuss the techniques of ontology engineering used to build such ontologies. We name the ontology as SemArtha, an acronym for Semantic ArthaShastra based on Kautilya's epic work on public administration. A dedicated OWL based ontology for E-Governance would enable semantic web based e-services especially, Government to Citizen(G2C) and Government to Business(G2B) services. Finally we analyze the working of a Java based module of SemArtha for electronic land registration.

Keywords — SemArtha, Ontology mediation, Ontology engineering, OWL

I. INTRODUCTION

Modern day electronic-government services have come a long way from the initial days of office automation and digitization of paperwork. Increasing scope of services provided by governments along with increasing expectations of the citizens for quick, efficient and transparent delivery of public services has transformed e-governance into a potent tool for the success of citizen centric services. In its 11th report the second Administrative Reforms Commission, set up by Government of India, emphasises on using e-governance to create SMART (Simple, Moral, Accountable, Responsive and Transparent) governance[1].

To achieve this, seamless integration of services offered by various government agencies is essential. This is where ontology based e-governance comes into the picture. An ontology is a form of knowledge representation that relates concepts within a domain. Formally, an ontology is defined as a four tuple $\langle C, R, I, A \rangle$ where C represents concepts, R represents relations, I represents instances and A represents axioms[2]. Use of ontologies in information processing increases machine understandability of the meaning

of data. This in turn helps in semantic interoperability among heterogeneous ontologies, thus allowing different applications to interact. Before heterogeneous ontologies, can communicate, their distinctions and mismatches should be resolved. This is done by ontology mediation.

In this paper, we propose a ontology based e-government public service delivery system named *SemArtha*. The name is an acronym for Semantic Arthashastra. We model the services provided by a town council (*Nagar Panchayat*). Major functions of a town council include collection of various local taxes like building tax, water tax etc, construction and maintenance of roads and bridges etc. Further, we also model the services provided by the government at the tehsil level, and show semantic interoperability between the town council and the tehsil. We use Web Ontology Language to represent ontologies[3].

II. BUILDING ONTOLOGIES

Ontologies can be created manually or by using semi-automatic methods[4]. We have used manual method for creation of ontologies. We have followed the following steps:

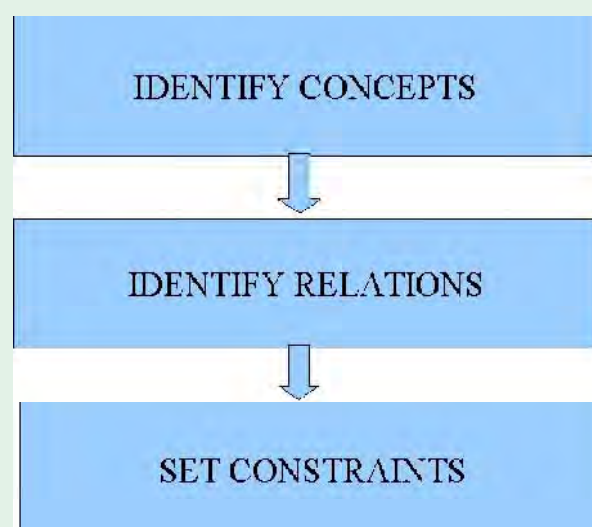


Fig. 1. Steps in manual ontology building

Step 1: Concept Identification: Major concepts are identified by taking help of domain experts and by analyzing existing laws. For example vehicle tax,

water tax, building construction tax and land tax are subclasses of the concept tax.

Step 2: Relations between identified concepts: In the second step all relations between identified concepts are established. For example vehicle tax and water tax are independent (disjoint) where as there exists a relationship between building construction tax and land tax.

Step 3: Constraint identification: Finally, the various constraints imposed by laws are identified and presented as axioms. For example the number of water connections to a building is limited by its land size and hence by land tax.

We use Protege[5], an opensource and free ontology editor to create and modify ontologies.

```
<rdf:RDF xml:base="http://www.odia-nlp.blogspot.com/np.owl">
<owl:Ontology rdf:about="">
<owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"> 1.0
</owl:versionInfo>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"> SemArtha by Utkal University</rdfs:comment>
</owl:Ontology>
<owl:Class rdf:ID="VehicleTax">
<rdfs:subClassOf>
<owl:Class rdf:about="#Tax"/>
</rdfs:subClassOf>
<owl:disjointWith>
<owl:Class rdf:about="#WaterTax"/>
</owl:disjointWith>
<owl:disjointWith>
<owl:Class rdf:about="#LandbuildingTax"/>
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="LandTax">
<rdfs:subClassOf>
<owl:Class rdf:about="#Tax"/>
</rdfs:subClassOf>
<owl:disjointWith>
<owl:Class rdf:about="#WaterTax"/>
</owl:disjointWith>
<owl:disjointWith>
<owl:Class rdf:about="#LandbuildingTax"/>
</owl:disjointWith>
</owl:Class>
```

Fig. 2. Ontology Representation using OWL

III. ONTOLOGY MEDIATION

Interoperability among different ontologies is central to the success of ontology based systems. Since all ontologies cannot be expected to follow same concepts and rules, differences in their representation needs to be resolved first. This resolution is known as ontology mediation. An ontology based system with efficient mediation techniques can increase cooperation between various government agencies. Ontology mediation can be achieved in many ways[6]. We use ontology mapping through Context-OWL(C-OWL)[7] for ontological mediation. C-OWL extends OWL by including representation for contextual ontologies. A contextual ontology is a local ontology that can be mapped to other ontologies through external mapping using bridge rules. OWL ontologies along with the mapping between them is called a context space. Fig 3 illustrates how C-OWL achieves semantic mediation.

```
<cowl:mapping>
<rdfs:comment>example of ontology mapping
</rdfs:comment> <cowl:sourceOntology
rdf:resource="http://www.Odianlp.blogspot.com/np.owl"/>
<cowl:targetOntology rdf:resource="http://www.Odianlp.blogspot.com/th.owl"/> <cowl:bridgeRule
cowl:br-type="equiv"> <cowl:sourceConcept
rdf:resource="http://www.Odianlp.blogspot.com/np.owl#WaterTax"/> <cowl:targetConcept
rdf:resource="http://www.Odianlp.blogspot.com/th.owl#JalKar"/> </cowl:bridgeRule>
<cowl:bridgeRule cowl:br-type="equiv">
<cowl:sourceConcept rdf:resource="http://www.Odianlp.blogspot.com/np.owl#LandTax"/>
<cowl:targetConcept rdf:resource="http://www.Odianlp.blogspot.com/th.owl#BhuKar"/> </cowl:bridgeRule>
<cowl:bridgeRule cowl:br-type="incompat"> <cowl:sourceConcept
rdf:resource="http://www.Odianlp.blogspot.com/np.owl#LandTax"/> <cowl:targetConcept
rdf:resource="http://www.Odianlp.blogspot.com/th.owl#JalKar"/> </cowl:bridgeRule>
<cowl:bridgeRule cowl:br-type="incompat">
<cowl:sourceConcept rdf:resource="http://www.Odianlp.blogspot.com/np.owl#WaterTax"/>
<cowl:targetConcept rdf:resource="http://www.Odianlp.blogspot.com/th.owl#BhuKar"/> </cowl:bridgeRule>
```

Fig. 3. Ontology Mapping using C-OWL

IV. SEMANTIC INTEROPERABILITY

Semantic interoperability is a natural outcome of semantic mediation. In SemArtha, semantic

interoperability is achieved through the use of ontological mapping by C-OWL. Payment of taxes in the town council and payment of land revenue at the tehsil level are two operations supported by two different ontologies. However, common concepts and relations also exist between the two ontologies. These commonalities and distinctions are mapped by C-OWL. We then use a java based module to provide various e-services. Interoperability is required in cases where both the ontologies have different representations for the same concepts. A simple example is the measurement of land by the town council in acres and by the tehsil in hectares. Other complex distinctions like difference in levels of abstraction in equivalent classes of different ontologies are also successfully dealt with.

V. CONCLUSION AND FUTURE WORK

Ontology based e-governance is rapidly gaining ground among researchers and practitioners in the field. Governments across the world have started implementing ontology based e-services for providing citizens with a seamless e-governance experience. Certain grey areas like information ownership and privacy of sensitive data still remain to be addressed.

SemArtha provides a complete ontology for providing e-services at the town council level. It can be extended to meet the requirements of larger self government units like municipalities and municipal councils. Current self government e-services are primarily data based and not concept based. As a result each self government unit maintains its own independent system though they do the same work. An ontology based approach to e-governance can improve efficiency and reduce wastage of time and space.

REFERENCES

- [1] 11th Report of the 2nd ARC, Government of India "Promoting e-Governance The SMART Way Forward"
- [2] Staab S, Studer R (Eds). 2004. Handbook on Ontologies. International Handbooks on Information Systems. Springer: ISBN 3-540-40834-7.
- [3] <http://www.w3.org/TR/owl-guide/>
- [4] Chapman P, Clinton J, Kerber R, Khabaza T, Reinartz T, Shearer C, Wirth R. 2000. CRISP-DM 1.0: Step-by-step data mining guide.
- [5] <http://protege.stanford.edu/>
- [6] Campbell, AE and Shapiro, SC, 1998, "Algorithms for ontological mediation" Technical Report 98-03, Department of Computer Science and Engineering, State University of New York at Buffalo.
- [7] Bouquet P, Giunchiglia F, van Harmelen F, Serafini L, Stuckenschmidt H. 2004. Contextualizing ontologies. *Journal of Web Semantics* 1(4):325.

Analysis of Ontology with Web Usage Mining in Semantic Web: An Overview

Sanjay Kumar Malik, Asst Professor

University School of Information Technology, GGS Indraprastha University, Delhi
sdmalik@hotmail.com

Abstract— Today, Internet is a huge database which contains a large number of Web sites, search engines and other information. Due to the unstructured and semi structured data in the web pages , it is a challenging task for researchers to make a fast , relevant and efficient search in warehouse of such type of database. Ontology may be a good mechanism for achieving this goal and Web Mining technique may be used to discover and extract meaningful information from the Web documents. In this paper, analysis of web usage mining along with ontology has been made with the help of an example of sample data for which “Web Log Expert” Log analyzer tool has been used and it has been appended with the development of an Ontology.

Keywords—Semantic Web, Ontology, Web Mining, Web Usage Mining, Web Log Expert, Web Log Analyzer

1. INTRODUCTION

The goal of Semantic Web is to make the data machine understandable in the form of Ontology which defines the structured web content while Web Mining semi-automatically extracts the useful knowledge or information hidden in these data or from World Wide Web (WWW) and making it available to the user. Web mining can be categorized into different classes based on which part of the web is to be mined where web usage mining plays a significant role in the discovery of meaningful patterns from the client-server transactions. For implementation the Web Usage Mining, Web Server Logs may be required where Web Log Analyzer may be used for pattern discovery and analysis.

1.1 Semantic Web

Semantic Web is about building an appropriate infrastructure for intelligent agents to run around the Web performing complex actions for their users (Hendler, 2001) and it is about explicitly declaring the knowledge embedded in many Web-based applications, integrating information in an intelligent way, providing semantic-based access to the Internet, and extracting information from texts [14]. Ontology is one of the layers of Semantic Web’s architecture as proposed by Sir Tim Berner’s Lee., which is used to make vocabulary for the system.

1.2 Ontology

Ontology is a domain knowledge that could be used to describe information on the Web. The term ontology can be defined in many different ways. Genesereth and Nilsson defined ontology as an explicit specification of a set of objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them [14]. Ontologies based web mining can be used to improve search to web data by adding Ontology annotations, better browsing capabilities and personalization of Web data from the user’s profile.

1.3 Web Mining

Web mining is the application of data mining techniques to extract knowledge from Web data.

Complexity of the web infrastructure and the focus on scalability has led to numerous data quality issues related to page view identification, visitor identification and robot activity filtering [2]. The qualities of the extracted interests are evaluated based on the criteria using the Page Rank Algorithms [5].

1.3.1 Web Mining Categories

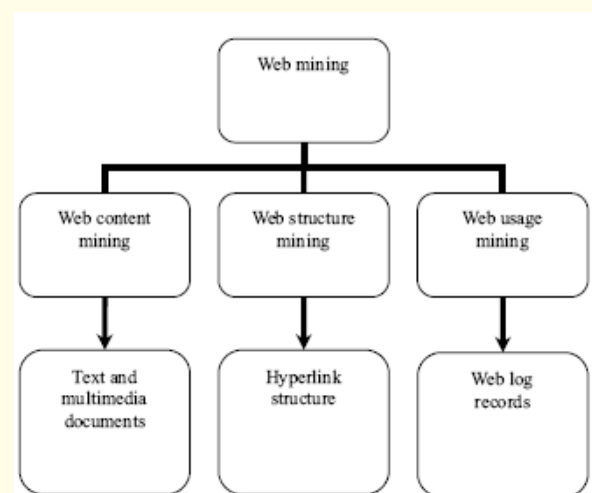


Fig. 1. Web Mining Categories [12]

The three web mining categories are as follows:

1.3.1.1 Web Content Mining

There are two types of approach in Web Content Mining, Database Approach and Agent Based Approach. Agent based Approach includes intelligent search agent, information filtering and categorization, and personalized web agent. Database approach includes Multilevel Databases and Web Query System [4]. Various techniques to extract data from Web mining exist like Web Crawler, Wrapper generation and Page Content Mining.

1.3.1.2 Web Structure Mining

Web Structure Mining works on the hyperlink structure of the web. The graph structure can provide information about ranking or authoritativeness and enhance search results of a page through filtering [13].

1.3.1.3 Web Usage Mining

Web Usage mining has been defined as the application of data mining techniques to large Web data repositories in order to extract usage patterns [3] and it is mainly used to general access pattern tracking and customize of usage tracking on the web server. Web Usage Mining may be achieved with the help of Web Server Log files. Log Files contain the behavior of visitor's interest and contains client IP address, access time, HTTP request method, path of the resource on the Web server, protocol used for the transmission., status code of the server and number of bytes transmitted in the transaction [11]. Mining applications are based on data collected from three main sources viz; Web Servers, Proxy Servers and Web Clients [8]. There are many kinds of data that can be available in Web Usage Mining viz; Content, Structure, Usage and user Profile [9]. Web usage mining is achieved first by reporting visitors traffic information based on Web server log files and other source of traffic data .In order to discover and analysis of usage patterns from the available data, it is necessary to perform three steps : Preprocessing, Pattern Discovery, Pattern Analysis [9]. Web mining technique is used to extract meaningful information from the Web document, where Ontology may be used to make structured document and therefore Ontology development is of prime significance. Below is the sample case study of the development of an Ontology.

2. ONTOLOGY DEVELOPMENT: A CASE STUDY

Here, Protégé version 3.4.1 tool is used to develop an Ontology on "University School of Law and Legal Studies".

2.1 "University School of Law and Legal Studies(SLLS)" Ontology

Following is the snapshot of SLLS Ontology in Protégé which shows relationship between different classes and sub classes. "Persons" and "Programmes" are two super classes of SLLS and "Staff" is sub class of "Persons" and then "Staff" inherits to "Non Teaching" and "Teaching" classes.

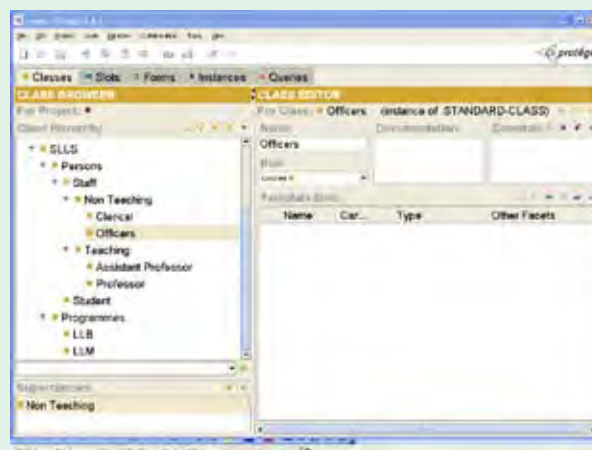


Fig. 2. "SLLS Ontology"

2.2 Code Snippets

In the process of development of Ontology, following code snippets are included:

2.2.1 OWL Code

Web Ontology Language(OWL)is used to standarize a more capable Ontology framework language in semantic web. It comes in three flavours viz, OWL Lite, OWL DL, and OWL Full [16]. Following is code snippet of OWL Full in the above Ontology:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns="http://www.owl-ontologies.com/Ontology1270838193.owl#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1270838193.owl">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="Student">
  <rdfs:subClassOf>
```


2.2.2 RDF Code

RDF, Resource Description Framework, is a flexible language capable of describing all sorts of information and meta data [16]. Following is code sippet of RDF in the said Ontology:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-
ns#'><!ENTITY a 'http://protege.stanford.edu/system#'>
<!ENTITY rdf_ 'http://protege.stanford.edu/rdf'>
<!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]><rdf:RDF xmlns:rdf="&rdf;"xmlns:rdf_="&rdf_;"
xmlns:a="&a;"xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&rdf_;Assistant_Professor"
rdfs:label="Assistant Professor">
<rdfs:subClassOfrdf:resource="&rdf_;Teaching"/>/
rdfs:Class>
```

2.2.3 XML Code

XML, Extensible Markup Language, is a language framework that has been used to define all new languages that are used to interchange data over the Web [16]. Following is code sippet of XML in the said Ontology:

```
<?xml version="1.0" ?>
<knowledge_base
xmlns="http://protege.stanford.edu/xml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://protege.stanford.edu/xml http://
protege.stanford.edu/xml/schema/protege.xsd">
<class>
<name>:SYSTEM-CLASS</name>
<type>:STANDARD-CLASS</type>
<own_slot_value><slot_reference>:ROLE</slot_reference>
<value value_type="string">Abstract</value>
</own_slot_value><superclass>:THING</superclass>
</class></knowledgebase>
```

A Web Log Analyzer may be used for pattern discovery and analysis in web usage mining.

3. WEB LOG ANALYZER

Web server log files were used by the webmasters and system administrators for the purposes of “how much traffic they are getting, how many requests fail, and what kind of errors are being generated”, etc. Analyzing and discovering Log could help to find more potential customers and trace service quality and so on [6]. There may be different types of Web Log Analyzer like Web Log Expert by Nihuo Software Company.

3.1. Web Log Expert: A Case Study

Web Log Expert is a fast and powerful access log analyzer. It gives information about site’s visitors: activity statistics, accessed files, paths through the site, information about referring pages, search engines, browsers, operating systems, and more. Web Log Expert can analyze logs of Apache and IIS web servers. It can read GZ and ZIP compressed log files so that we won’t need to unpack them manually [10].For implementing Web Usage Mining, Web Server Logs may be required. Here, following sample Web Logs which has been provided by the Web Log Expert Tool has been used as an example for analysis:

```
63.238.163.75 - - [30/Aug/2004:03:56:58 +0000] "HEAD
/ HTTP/1.1" 200 0 "-" "InternetSeer.com" coolfilesearch.
com text/html "/usr/home/us61q1/htdocs/index.html"
```

```
195.159.181.230 - - [30/Aug/2004:03:59:46 +0000] "GET
/images/glass.gif HTTP/1.1" 200 990 "http://download-
soft.com/Catalog/Most-Popular/Utilities/File&Disk-
Management/catalog4.htm" "Opera/7.54 (Windows ME;
U) [nb]" coolfilesearch.com image/gif "/usr/home/us61q1/
htdocs/images/glass.gif"
```

Fig. 3 “Sample Web Server Logs – Sample Data by Web Log Expert”

With the use of these Logs, Web Log Expert generates some of the following reports about the visitor’s information and their interest as in figure 4, 5, 6:

3.1.2 Experimental Results

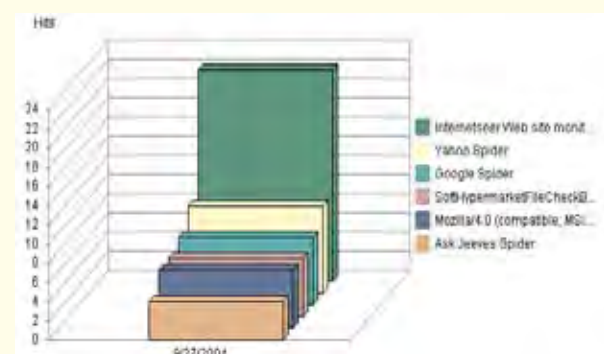


Fig. 4. “Report – “Daily Spider Activity”

Figure 4, depicts the activity of search engine to search website from the World Wide Web, Figure 5, refers to rating of the different web browser to searching and Figure 6, refers the report of error pages on the website.

4. ONTOLOGY AND WEB USAGE MINING

Web Usage Mining refers to the discovery of knowledge from Web Server which includes Web pages, Web links, and Web log data. Figure 7, relates Semantic Web, Ontology and Web Usage Mining. The traditional topics covered by Web content mining include Web clustering, Web page classification and Web extraction where Ontologies may be applied as background semantic structures for Web mining. Using Ontology within the Web usage mining process or indeed during the deployment of web usage mining remains a challenge [2].

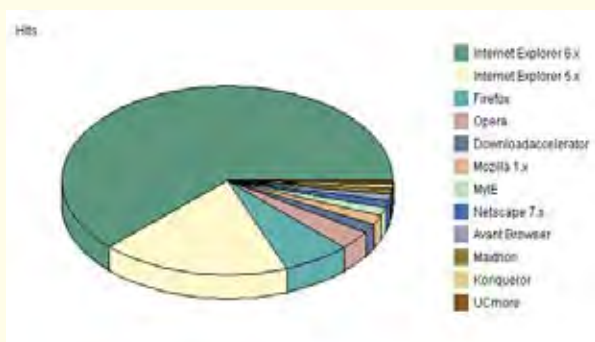


Fig. 5. "Report – "Top Browser"

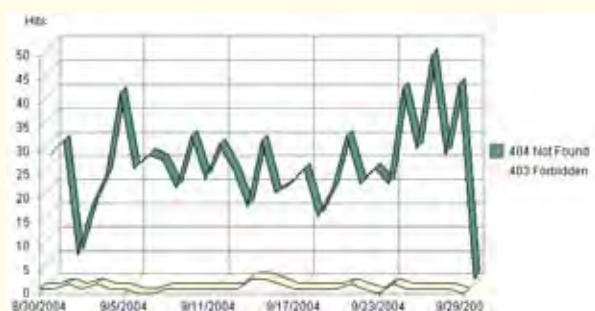


Fig. 6. "Report – Error"

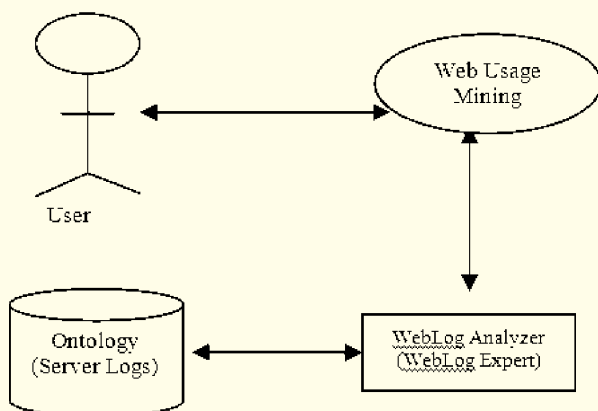


Fig. 7. "Relating Ontology with Web Usage Mining"

5. CONCLUSIONS AND FUTURE WORK

Web usage mining can be used in Counter Terrorism, Fraud Detection, and detection of unusual accesses to secure data. Web Usage Mining may be referred to make search relevant by determining frequent access behavior for users, needed links can be identified to improve the overall performance of future accesses. This paper presents an analysis of the development of the structured data in the form of Ontology and Web Mining for extracting the useful knowledge hidden in the data and the usage of a Web Log Analyzer for pattern discovery and analysis.

6. REFERENCES

- [1] Mobasher, B., R. Cooley and J. Srivastava Data Preparation for Mining World Wide Web Browsing Patterns, in the Journal of Knowledge and Information Systems, Vol. 1, No. 1,1999
- [2] S. Singh Anand, M. Mulvenna, and K. Chevalier, "On the Deployment of Web Usage Mining", EWMF 2003, LNAI
- [3] Cooley, R., Tan, P-N., Srivastava, J. Discovery of Interesting Usage Patterns from Web Data, Web Usage Analysis and User Profiling, In Lecture Notes in Artificial Intelligence, Brij Masand, Myra Spiliopoulou (Eds.), pp. 163-182, 2000.
- [4] WangBin, LiuZhijing, "Web Mining Research ", Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03), IEEE
- [5] Page, L., Brin, S., Motwani, R., Winograd, T.,The PageRank Citation Ranking: Bringing Order to the Web,http://www-b.stanford.edu/~backrub /pageranksub.ps,1998.
- [6] Hong T, Chiang M, Wang S H, "Mining weighted browsing patterns with linguistic minimum supports", 2002 IEEE International Conference on Systems, Man and Cybernetics, 2002,Yasmine Hammamet, Tunisia, pp. 635-639.
- [7] Qingtian Han, Xiaoyan Gao, "Research of Distributed Algorithm Based on Usage Mining", 978-0-7695-3543-2/09, 2009 IEEE
- [8] R. Cooley, B. Mobasher and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November.
- [9] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", ACM, SIGKDD, Jan 2000

- [10] Nihuo Software, web log analyzer, <http://www.loganalyzer.net>
- [11] R.M. Suresh, R.Padnajavalli, "An Overview of Data Processing in Data and Web Usage Mining" 2006 IEEE
- [12] Miguel Gomes da Costa Júnior, Zhiguo Gong, "Web Structure Mining: An Introduction", Proceedings of the 2005 IEEE International Conference on Information Acquisition, June 27 - July 3, 2005, Hong Kong and Macau, China
- [13] A.Senthil Kumar, N.Palanisamy, "Challenges for Web Mining", Proceedings of the 2008 International Conference on Computing, Communication and Networking (ICCCN 2008), 2008, IEEE
- [14] R.M. Suresh, "A Study on the Ontology Based Web Mining For Digital Library", IET-UK International Conference on Information and Communication Technology in Electrical Sciences, ICTES 2007
- [15] Vladan Devedzic, University of Belgrade Serbia and Montenegro, "Semantic Web and Education", Springer, pp 38-43
- [16] Thomas B. passin, "Exploring Guide to the Semantic Web", pp-33, pp-161

Unstructured Knowledge Representation using Polyscopic Modelling

Dongre Rohit V., P. Santhi Thilagam

Department of Computer Engineering,
National Institute of Technology Karnataka Surathkal, India - 575025
rohiitvdongre@gmail.com, santhi_soci@yahoo.com

Abstract— Knowledge management focuses on preserving knowledge gathered from past experiences and make it available for future references. As quantity of information to be managed is increasing knowledge management is becoming crucial. Most of such information sources like Wiki, blogs, documentations are unstructured. To acquire knowledge from these unstructured sources, a machine readable knowledge representation is required. Previously bag-of-words approach was used for representing such documents. But they lack in representing semantic knowledge hidden in unstructured documents. We generate a meta-data layer which stores semantic knowledge extracted from unstructured documents. This will help to refine knowledge retrieval process from unstructured documents.

I INTRODUCTION

The rapid growth of the Internet has resulted in an enormous number of heterogeneous and unstructured documents. Large part of knowledge is stored in these textual documents. This knowledge cannot be queried and accessed in a straight forward way. We need a formal way to represent this unstructured textual knowledge in such a way that it can be effectively access and reused. In this scenario, knowledge management highlights a need for a knowledge management systems which can transform textual knowledge in a machine readable form. A sophisticated and usually additive approach is to overlay unstructured resources with meta-data (superimposed information) in order to refine retrieval process.

A well known knowledge discovery method, text mining, focuses on discovering new knowledge such as trends and patterns hidden in huge collection. One of the application would be to find relation between two documents. Such kind of association will extract hidden knowledge from the textual sources.

But analyzing unstructured knowledge sources is not possible without representing them as a formal structure. Representation mechanism used for storing textual knowledge is crucial for its efficient usage. Because of this, knowledge representation plays an important role in knowledge management. Various

techniques have been used for knowledge representation. Broadly these mechanism can be classified into two categories. Representations storing semantics within document or not. With huge information sources, knowledge representation techniques which are not based on semantics or contextual information are not efficient. Semantic based methods like Topic Maps, Conceptual Graphs, Semantic Networks, Ontologies are advantageous for representing unstructured knowledge. They represent knowledge by using relations occurring in unstructured document by forming meta-data layer over base data.

In our approach, we use Conceptual Graphs[1] to transform unstructured documents into machine readable format. Conceptual graphs forms a meta-data layer superimposed over unstructured data. Conceptual graph is a network of concept nodes and relation nodes. In conceptual graph, concepts occurring in the document are stored as a concept nodes and relations by which concepts are related are represented by relation nodes. Identification of these relations from unstructured documents is a primary step for conceptual graph generation. Extracting these relations from document can be done in two ways, namely supervised or unsupervised extraction. In supervised relation extraction target patterns are given as input to relation extractor. Only such patterns are searched in text and extracted. On the other hand in unsupervised relation extraction process, no initial patterns are determined. Context information, in which basic elements can occur, is used for extracting relations. We follow context based relation extraction strategy for relation identification and extraction. Generalized grammar rules in English language are used for providing contextual information for the extraction process.

In this work we use simplified notion of conceptual graph as a meta-data for representing unstructured document. Conceptual graph for sentence “Sam is eating food” is shown in Figure 1.

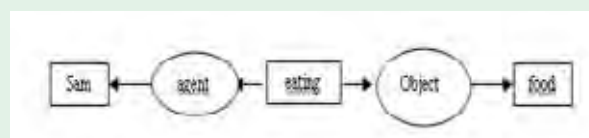


Fig. 1. Conceptual Graph

Organization of information sources in a way that it makes resources easily accessible is equally important once knowledge representation is done. With enormous increase in knowledge sources, well structured organization of sources is very important. Polyscopic modeling [8] terminology helps to organize sources in a better way. It distinguishes high level and low level views of the same information. Polyscopic modelling focuses on providing level based view of the information by distinguishing between abstract and detailed view. This level based information structuring provides better categorization of the base data. Polyscopic structuring of information helps to generate user specific view of the same data.

We concentrate on representing unstructured knowledge into a graphical structure by using conceptual graphs. This graphical structure is built in a way that will facilitate polyscopic organization of knowledge sources.

II. RELATED WORK

Information extraction from the unstructured sources has been always an area of interest for researchers. Generation of superimposed meta-data layer over unstructured data is good approach. To handle unstructured information, Matteo M. and Danilo M.[11] proposed logical data model based on the unstructured data. Initially information extraction from the unstructured sources was based on the bag-of-words or term vector methodology. David W. et al.[5] transformed the unstructured data to the structured information using ontology by extracting keywords and constants from the document. Term vector based methods also suffer from high dimensionality factor. Hotho et al.[3] used vector of concepts in place of term vector, which reduces the dimensionality of the vector considerably. Such bag-of-words approach lags in representing contextual/semantic information.

Bernotas et al.[12] used tagging approach to extract information in association with the ontologies. Each document is represented as a multi-dimensional vector encompassing not only words appearing in document but also concepts, which are extracted by using ontology.

High dimensional term vector approach do not consider the semantics of the context in which word is used. Thus we need a method that can provide multiple subjective perspectives on the same document set[3]. This drawback is avoided by extracting semantic relations from the unstructured documents.

To extract semantic relations Rozenfeld et al.[4] described relation discovery by clustering frequently

co-occurring pairs of entities based on context in which they appear. But here they only consider co-occurrence of words, which is not a complete semantic information. Khaled Hammouda and Mohamed Kamel [7] used phrase based representation.

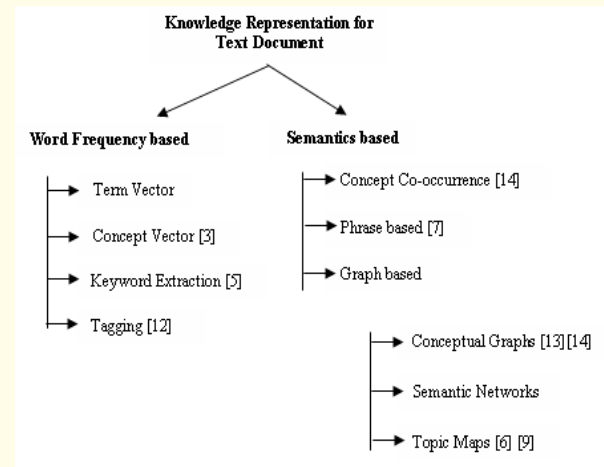


Fig. 2 Text Document Representation Taxonomy

To overcome drawback of vector based methods, graph based methods are proposed. Karabeg et al.[6] and Knud et al. [9] used topic maps for structuring information polyscopically. Karabeg et al.[6] used polyscopic modelling in university course model. But in this approach each document has to be explicitly associated with relevant information. Major drawback of topic map is standard topic maps do not provide formally defined concepts. Also structuring of contents depends on ability of the author.

Tao et al.[13] used conceptual graphs as a representation of the unstructured information. Conceptual graphs are then processed to infer the knowledge from text. But they extracted fixed patterns from the documents. ConKMeI [14] used conceptual graphs for knowledge representation. Also discussed about hierarchical knowledge representation.

Polyscopic [8] modeling is used for contextual organization of information. Polyscopy employs conscious creation of multiple scopes and views. Depending on the scope of the viewer, information is provided to the user. We propose to combine Polyscopic modelling approach to conceptual graphs for representing knowledge from unstructured documents.

III PROBLEM DESCRIPTION

Unstructured documents $d_1, d_2, d_3, \dots, d_n$ forms a raw source of knowledge, where d_i represents the i^{th} unstructured document. Let $D, D = \{d_1, d_2, \dots, d_n\}$

be the set of all unstructured documents d_i , where $i = 1 \dots n$. This unstructured document set D is an input to first phase. Some machine readable intermediate representation is necessary to extract knowledge from unstructured document d_i . We address the problem of representing each document d_i in set D by building superimposed layer, using conceptual graphs. To generate conceptual graphs we extract relations from each unstructured document d_i . Relations are extracted in first phase and given as input to second phase. Document d_i is represented by a conceptual graph g_i , which is generated as an output of second phase. Each conceptual graph, g_i , is a bipartite graph, whose nodes are divided into two disjoint sets concept node (C) and relation nodes (R), such that every edge connects a vertex in C to one in R; that is, U and V are independent sets. Suppose P is a set of conceptual graph g_i , thus $P = \{g_1, g_2, \dots, g_n\}$, contains output conceptual graph for each document d_i in input set D . We also focus on organizing these conceptual graphs in set C in an hierarchical structure.

IV. SOLUTION APPROACH

Unstructured documents are organized into logical structure through three phase processing. In first phase we identify and extract relations from an unstructured document. After generation of relations, respective conceptual graphs are generated, which are used as an image of unstructured document for further processing. Finally, we organise documents using this conceptual graph representations. Figure 3 shows overall view of solution approach.

A. Relation Identification and Extraction

Pre-processing: Unstructured documents contains information as a raw text. Structured information sources like relational database, stores information which can be easily understood. In preprocessing phase, we identify sentences within a document. This is important, since conceptual graph generation is done at sentence level. Relations are extracted firstly for a sentence to generate conceptual graphs for it. These conceptual graphs are further combined to form conceptual graph for a document. In this step we also remove punctuation marks and other special symbols from text.

Each sentence is then parsed by a opennlp Treebank parser[2] to generate sentence grammar tree. Each sentence consists of various basic elements. Parser generates a sentence parse tree and assigns each element appropriate part of speech (POS) tag/word class. POS tag determines the role of that word in the sentence. POS tag assigned to word may depend on the context in which word appears. This ambiguity is resolved by

parser with the help of pre-defined set of rules and using probabilistic parsing with the help of training data.

Output of this step is a tagged sentence parse tree. This is given as input to relation extraction phase. For example, consider sentence “Red roses are a pretty ornament for a party” is present in a document. After this phase, each word in the sentence is associated with POS tag, ‘Red’ is an adjective, ‘roses’ is a noun, ‘are’ is a verb etc. This tag information is used by next phase.

Context based Relation Extraction: Relation extraction from a sentence is the most important step for meta-data generation process. These extracted relations are be stored as a meta-data using conceptual graphs. Sentence tree with tagged nodes only signifies the class of particular word in the sentence. But what role is played in the sence depends on context of occurrence. For example a Noun can be a subject or object in sence depending on the context in which it is present. Based on the tag attached to word and its context information we identify role of that word in sentence.

This process is sub-divided into Clause determination, Clause fragmentation and finally extracting relations

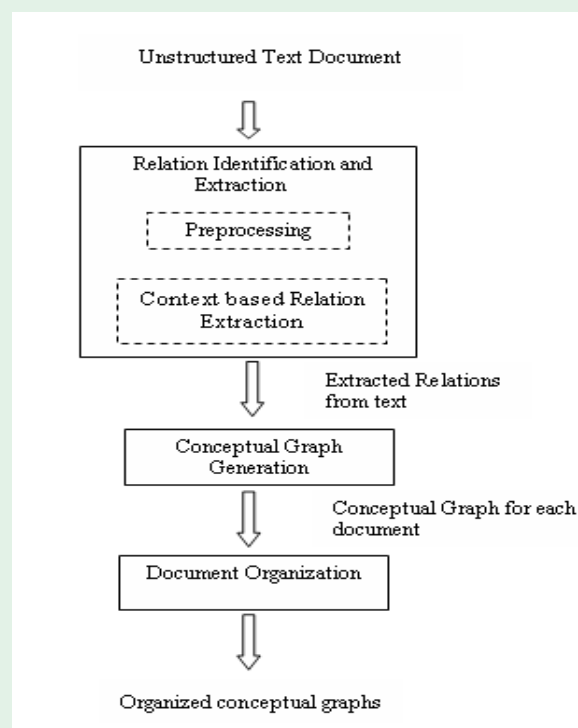


Fig. 3. Unstructured Documents Processing Flow

1. **Clause Determination:** Each sentence in a English language can be analyzed as a hierarchical structure. Clause is a syntactic unit consisting of a verb together

with its associated subject, object or complements and adverbials. Any clause must have the subject and the verb, may not be in same order. Because of this, each sentence is initially sub-divided into clause.

Most of the conceptual graph papers focus on extracting some patterns from documents. But we focus on extracting as much relations from documents as possible. More extracted relation in CG will cause more effective representation of document in meta-data layer.

2. Clause Fragmentation: Clause consists of basic elements like subject, object, verb etc. We have to identify these basic elements from each clause. For making this identification process easier, we divide each clause in three fragments. This helps in determining indirect relationships between basic sentence elements. Three fragments can be identified as follows.

First Fragment : Part of a clause which occurs before verb/verb associated elements, is identified as First Fragment.

Verb Fragment : Verb is obligatory part of any clause. Verb fragment consists of verb and other supporting elements to verb. Supporting element for verb may be adverb, supporting verb, participles etc.

Remaining Fragment : Sentence elements occurring after verb fragment are identified as remaining fragment.

After fragmentation of clause, relations between basic elements occurring in a fragment are extracted. Relations between inter-fragment elements are identified parallelly. We identify relations like subject, object, attribute for clause elements.

In the above example sentence, we have noun 'roses'. 'Red' is an adjective for roses, between these entities there exists an 'attribute' relation. Similarly in sentence, 'roses' is a subject of verb 'are' and 'ornament' is an object of verb. 'pretty' is an adjective for 'ornament', they are also related by attribute relation. In this way we extract possible relations from sentences with the help of contextual information. These extracted relations are next transformed into conceptual graphs.

B. Conceptual Graph Generation:

Conceptual graph generation process starts with generation of conceptual graph for relations extracted from a sentence.

Process of conceptual graph generation for a

document combines conceptual graphs for each sentence.

Conceptual graph generation of sentence starts from cg generation for a clause fragment. After generating conceptual graph for each clause fragment, these individual conceptual graphs are combined to represent CG for a sentence. This process completes conceptual graph generation for a sentence.

In the process of generating conceptual graph for a document, redundant graph entries for already present relations or concept nodes are avoided. If node is already present in a conceptual graph then node is not inserted again. Also relation entries made in the graph are case insensitive. This means, concept/relation nodes in graph are not distinguished by case in which they appear in text. We do not consider *determiners* as a concept node for insertion in conceptual graph. While traversing a verb fragment, we consider only main verb, neglecting all supporting verbs from the fragment.

Figure 4 shows conceptual graph for above example sentence. In the conceptual graph concept nodes are represented in a square boxes and relational nodes are represented in ovals.

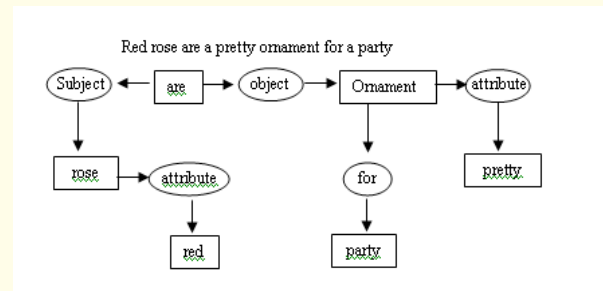


Fig. 4. Conceptual Graph

C. Document Organization

We group similar documents into clusters. Clustering is done by using graph similarity measure[10]. But, just graph similarity measure is not sufficient, we also need a sub-graph similarity measure. This situation arises when one document is a sub-document of the other. Similarity measure only gives the degree of similarity of documents, but it do not specify whether one document is contained by other or not. For this reason we use a sub-graphs similarity measure. So, finally we used similarity measure as a combination of graph similarity measure and sub-graph similarity measure.

Comparing two graphs is based on overlap graph of two input graphs. The overlap graph G_c of the two initial conceptual graphs G_1 and G_2 consists of the following elements:

- All concept nodes that appear in both initial conceptual graphs G_1 and G_2 .
- All relation nodes that appear in both G_1 and G_2 and relate the same concept nodes.

Figure 5 shows two sample conceptual graphs G_1 and G_2 . The overlap graph G_c is a set of all maximal common subgraphs of G_1 and G_2 as shown in Figure 6.

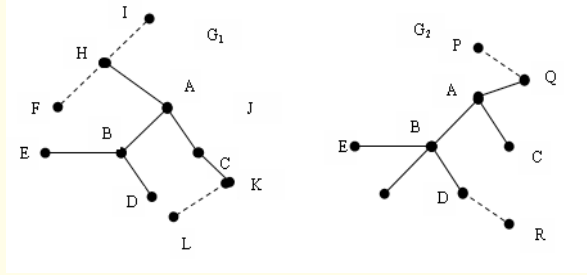


Fig. 5. Graphs G_1 and G_2

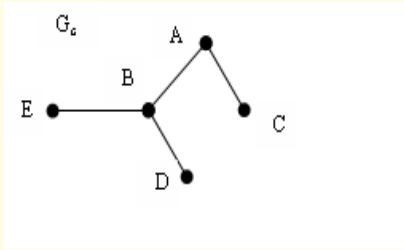


Fig. 6. Overlap graph G_c of G_1 and G_2

After generating overlap of two graphs, graph similarity is measured as a relative size of their overlap.

For conceptual graphs G_1 and G_2 and the overlap graph $G_c = G_1 \cap G_2$, the similarity, S_{graph} , between them is calculated as a combination of two values: conceptual similarity (S_c) and relational similarity (S_r).

The similarity measure is a value between 0 and 1, where 0 indicates that there is no similarity between the two pieces of knowledge, and 1 indicates that they are completely similar. Because of the bipartite (concepts and relations) nature of the conceptual graph representations, the similarity measure is defined as a combination of two types of similarity: the conceptual similarity (S_c) and the relational similarity (S_r).

Conceptual similarity measures how similar the concepts and actions mentioned in both pieces of knowledge are.

Relational similarity measures the degree of similarity of the information about these concepts (concept interrelations) contained in the two pieces of

knowledge. It indicates how similar is the context of the common concepts. Concept Similarity is defined as,

$$S_c = \frac{2n(G_c)}{n(G_1) + n(G_2)} \quad (1)$$

Where $n(G)$ is the number of concept nodes of a graph G .

Relational Similarity is defined as,

$$S_r = \frac{2m(G_c)}{m_{G_c}(G_1) + m_{G_c}(G_2)} \quad (2)$$

Where $m(G_c)$ is the number of arcs in the graph G_c and $m_{G_c}(G)$ is the number of the arcs in the immediate neighborhood of the graph G_c in graph G . The immediate neighborhood of G_c in G consists of the arcs of G with at least one end belonging to G_c .

Two components of the graph similarity, S_c and S_r , are calculated as above. These two measures are combined into a cumulative measure S_{graph} . Cumulative graph similarity measure is calculated as follows

$$S_{\text{graph}} = S_c \times (a + b \times S_r) \quad (3)$$

Values of coefficients a and b depends on the structure of the graphs G_1 and G_2 . Parameter a and b is given by

$$a = \frac{2 \times n(G_c)}{2 \times n(G_c) + m_{G_c}(G_1) + m_{G_c}(G_2)} \quad (4)$$

$$b = 1 - a \quad (5)$$

Sub-graph similarity for graphs G_1, G_2 with overlap graph G_c is calculated as

$$S_{\text{subgraph}} = n(G_c) / \min(n(G_1), n(G_2)) \quad (6)$$

So, final similarity measure is given by

$$S_{\text{final}} = S_{\text{graph}} \times S_{\text{subgraph}} \quad (7)$$

As documents are clustered, a representative conceptual graph (RCG) is generated for each cluster.

RCG works as a overall image of documents belonging to respective cluster. We use supervised clustering with training data. From the set of input documents, we randomly select some documents and use them to train initial clusters. After training procedure, remaining documents are clustered. The training process generates RCG for each cluster. When we want to insert new document in the cluster, we compare it with only cluster RCG's. Comparison of new document with all documents belonging to the cluster is not required. Document is added to the cluster in which we get maximum similarity measure.

When new document is added to existing cluster, RCG for respective cluster is updated. Updation procedure checks relations present in new conceptual graph and RCG. If there exists some relation which are present in new document and not present in RCG then such relations are added into RCG. So this updated RCG now contain information about newly added document.

After clustering of conceptual graphs, we organise documents within cluster in a hierarchical structure. This is purely based on similarity of each document in the cluster with corresponding RCG. For this polyscopic organization of documents, after completion of clustering process, similarity score for each document in a cluster with RCG is calculate again. This is done mainly to find multi-level hierarchy of documents within cluster. Each level indicates the set of documents which are almost equally similar to RCG. This multi-level organization helps to determine the depth of information content in the document. Once level structure for documents within cluster is found we try to find inter-level relations between document. Namely we try to find sub-graph relation between documents. This helps in concluding whether the document is a subset of other document in the cluster. This leads to discover any abstract and detailed level views for a same document.

V. EXPERIMENTAL STUDY

We conducted our experiments on a set of news articles. We collected news articles on topics Education, Cricket, Technology Terrorism, India. Each document is processed using abovementioned framework. A meta-data in the form of conceptual graph is generated for each document. These conceptual graphs are used for clustering. Generation of conceptual graph for a unstructured document is a one time process. We store generated conceptual graph for each document in a file.

We randomly selected some articles on each topic and used these for initial training of clusters. After

completion of training process, clustering process is done for all input documents which are represented as conceptual graph. In the clustering process, once training is done, document to be clustered is compared with representative cluster graph for a respective cluster. By following this process we avoid comparing each document with every other document for clustering.

Within cluster we tried to found hierarchical relationships between documents. We try to find documents which are subset of other documents in a cluster. After clustering we compared documents within cluster and we are able to find such relationships.

VI. CONCLUSION

Framework described above enables handling unstructured information sources by generating a logical meta-data layer. Conceptual graphs are used as a intermediate representation for unstructured information sources. We followed generic English language syntax for extracting relations from text. Conceptual graphs representation for an unstructured information effectively stores semantic knowledge unlike other systems based on bag-of-words scheme.

REFERENCES

- [1] Conceptual graph standard, http://www.jfsowa.com/cg/cg_standw.htm.
- [2] Opennlp tool, <http://opennlp.sourceforge.net>.
- [3] A.M. A. Hotho and S. Staab. *Ontology-based text document clustering*. <http://www.jfsowa.com/cg/cgstandw.htm>, 2002.
- [4] R. F. Benjamin Rozenfeld. *High-performance unsupervised relation extraction from large corpora*. In Proceedings of the 6th International Conference on Data Mining, 2006.
- [5] M. C. R. D. S. David W., Embley Douglas. *Ontology-based extraction and structuring of information from data-rich unstructured documents*. *CIKM*, pages 52–59, 1998.
- [6] R. G. Dino Karabeg and T. W. Nordeng. *Flexible and exploratory learning by polyscopic topic maps*. In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (*ICALT'05*), pages 946–947, 2005.
- [7] K. M. Hammouda and M. S. Kamel. *Document similarity using a phrase indexing graph model*. *Knowledge and Information Systems*, 6(6):710 – 727, 2004.
- [8] A. Karabeg and D. Karabeg. *Polyscopy - A new paradigm in design for the web*. In Proceedings of the 9th International Conference on Information Visualization, 2005.

- [9] W. E. Knud Steiner, Roland Wagner. *Topic maps - an enabling technology for knowledge management*. In 12th International Workshop on Database and Expert Systems Applications, 2001.
- [10] A. F. G. M Montes-y Gomez and A. Lpez. *Comparison of conceptual graphs*. *MICAI 2000: Advances in Artificial Intelligence .Lecture Notes in Artificial Intelligence N 1793*, Springer-Verlag, pages 548–556, 2000.
- [11] M. Magnani and D. Montesi. *A unified approach to structured, semistructured and unstructured data*. Technical Report UBLCS-2004-9, Department of Computer Science, 2004.
- [12] R. L. A. S. Marijus Bernotas, Kazys Karklius. *The peculiarities of the text document representation, using ontology and Tagging-based clustering technique*. *Information Technology And Control*, 36(2):217– 220, 2007.
- [13] A.-H. T. Tao Jiang and K.Wang. *Mining generalized associations of semantic relations from textual web content*. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(2):164–179, 2007.
- [14] A. M. Weihong Huang, Mike O’ Dea. Conkmel: A contextual knowledge management framework to support intelligent multimedia e-learning. In *Proceedings of the IEEE 5th International Symposium on Multimedia Software Engineering (ISMSE’03)*, pages 223–230, 2003.

Ontology-Driven Approach to Extract Domain Specific Chunks for Natural Language Enabled Enterprise Applications

Shailly Goyal¹, Shailja Gulati²

Innovation Labs, Tata Consultancy Services Ltd Gurgaon, India

¹shailly.goyal@tcs.com ²shailja.gulati@tcs.com

Abstract— For a robust natural language (NL) interface for enterprise application systems, understanding the exact semantics of users' NL query is a crucial and complicated task. The complexity arises due to the inherent ambiguity in natural language which may result in multiple interpretations of the user's query. We employ semantic web technologies to resolve this ambiguity. A part of the domain knowledge -application data and the meta-information about the application data is expressed as domain ontology. A general purpose NL parser is first used to obtain all the syntactically correct parses of the user's query. The syntactic parse which is consistent with the domain ontology is used to identify the domain specific chunks of the users query. These chunks help in understanding the intent of the input query, and assist in its answer extraction. Our approach seamlessly works across various domains, provided the corresponding domain ontology is available.

Keywords– Natural language interfaces, ontology, domain-specific chunking.

I. INTRODUCTION

Natural language (NL) enabled question answering systems to business applications [1], [2] aim at providing appropriate answers to the user queries. In such systems, query interpretation is a fundamental task. However, due to the innately ambiguous nature of the natural language, interpretation of a user's query is usually not straightforward. In order to resolve such ambiguities, NL enabled question answering systems mostly use general purpose NL parsers. Although these parsers give syntactically correct chunks for a sentence, these chunks might not be semantically meaningful in a domain. Thus the chunks obtained from such parsers may not be helpful in extracting the answer to the user's query. The problem becomes even more severe in case of complex queries involving multiple constraints and nested sub-questions. Thus the problem at hand is "How can we automatically enrich the output of a general purpose NL parser with the domain knowledge in order to obtain syntactically as well as semantically valid chunks for the queries in the domain?"

We put forward an approach to solve the queries in a domain using the domain knowledge and the syntactic

structure of the queries. A part of the domain knowledge is represented in the form of domain ontology which is based on semantic web technologies [3]. We define 'constraints' and 'semantically valid chunks' for a query. Semantically valid chunks aid in the interpretation and extraction of the answers to the user's queries unambiguously.



Fig. 1. The Architecture

II. THE ARCHITECTURE

NL based Question Answering system requires the queries to be analyzed and chunked in an appropriate manner so as to have correct query generation and answer extraction. An NL query can be viewed as consisting of a set of *unknown predicates* whose values need to be determined based on the *constraints* imposed by the rest of the query. Domain ontology along with a POS tagger is used to identify the constraints in the query. These constraints along with the domain knowledge and the parse structure of the query are used to find the semantically valid chunk set. These chunks are then converted to a formal query language and the answer is retrieved from the ontology. Figure 1 demonstrates the overall architecture of our approach. In this figure, solid arrows represent the process flow for a query, and dashed arrows represent the information flow.

Hence for the appropriate interpretation and analysis of the query, the important issues that need to be addressed can be summarized as:

- Constraint identification:* This involves identifying the correct predicate-object pairs.
- Semantically valid chunk set:* This involves identification of valid constraints for each unknown predicate so that correct interpretation of the given query can be ensured.
- Query generation:* In this step, the semantically valid chunk set is converted to appropriate formal language query using the domain ontology.

In Section III we briefly describe the domain ontology and formalize the terminologies used in this work. In the subsequent sections we discuss constraint identification and semantically valid chunk set formation.

III. DOMAIN ONTOLOGY AND OTHER PRELIMINARIES

We use semantic web technologies [3] to create the domain ontology (in RDF format) using the relational data of the business application along with its meta information stored in the seed ontology¹ [4]. The ontology DO of a domain D describes the domain terms and their relationships in the *subject .. predicate .. object* format. For illustration, Ritesh -project name -Bechtel describes that the predicate 'project name' of the subject 'Ritesh' has object 'Bechtel'. A synonym dictionary having information about the synonyms of the domain terms is also maintained.

The domain ontology and synonym dictionary are used to identify the concepts in the user query Q posed in the domain D . The domain ontology D_o is used to further classify the concepts as predicates and objects. For a query Q , we denote the *set of predicates* as $PQ = \langle p_1, p_2, \dots, p_n \rangle$ s.t. $p_i \in DO$, and p_i is present in the query Q . The *set of objects* present in the query Q is $OQ = \langle o_1, o_2, \dots, o_m \rangle$ s.t. $p_i \in DO$ or o_i is a numerical/date value, and o_i is present in the query Q .

For illustration, consider the query:

Example 1. What is the role of the associates who joined before 18/10/2008 in the SWON projects with costing and revenue more than \$15000 and < \$25000, respectively?

In this query, the concepts identified using the domain ontology are: role, employee name², joining

date, SWON, project name, costing, and revenue. After considering the date and the numeric values in the query, i.e., 18/10/2008, 15000 and 25000 as objects, the predicate and object set obtained are:

$PQ = \langle \text{role, employee name, joining date, project name, costing, revenue} \rangle$,

$OQ = \langle \text{SWON, 18/10/2008, 15000, 25000} \rangle$.

In the following section we discuss constraints identification of the given query.

IV. CONSTRAINT IDENTIFICATION

For a successful query creation and execution, identification and formulation of correct constraints is of utmost importance. With reference to this work, constraint identification involves binding each '*object*' in the query with its corresponding '*predicate*'. This predicate-object pair is referred to as '*constraint*'.

We define a *constraint* as $ck = (p_i, o_j)$, $o_j \in OQ$, $p_i \in PQ$, and o_j is the value for the predicate p_i in Q . All the constraints in the query Q are identified, and $CQ = \langle c_1, c_2, \dots, c_m \rangle$ denotes the *constraints set*. Predicate used in any constraint is referred to as *constraint predicate*. The *set of constraint predicate* is $P_Q^C = \{p_i \mid p_i \in PQ \text{ such that } \exists (p_i, o_i) \in CQ\}$. Predicates that do not form part of the constraint set are referred to as *unknown predicates*. The *set of unknown predicates* is $P_Q^U = \{p_i \mid p_i \in PQ, p_i \notin P_Q^C\}$.

For illustration, for Example 1 we obtain the constraint set, constraint predicate set and the unknown predicate set as:

- $CQ = \{(\text{joining date, } < 18/10/2008), (\text{project type, SWON}), (\text{costing, } > \$15000), (\text{revenue, } < \$25000)\}$.
- $P_Q^C = \{\text{joining date, project type, costing, revenue}\}$.
- $P_Q^U = \{\text{role, employee name, project name}\}$.

The constraint sets thus obtained are used to find the semantically valid chunk set as discussed below.

¹The seed ontology has meta information about the domain, like `ns:employee owl:type owl:person`, `ns:employee ns:hasName ns:employee name`.

²The synonym dictionary has mappings defined between 'employee' and 'associate' etc.

V. SEMANTICALLY VALID CHUNK SET

Semantically valid chunk set identifies the conditions on each unknown predicate in the query, and are constituted from the constraints and unknown predicates as obtained in Section IV. For instance, in Example 1 (Section III), ‘joining date < 18/10/2008’ is a condition on the predicate ‘employee name’. Due to the syntactic ambiguity, more than one syntactic parse might be obtained for a NL query. Such cases may eventually result in more than one semantically viable chunk set. Formally we define semantically viable chunk sets as follows. **Definition.** A *Semantically viable chunk set* (SVC set) of a query Q corresponding to the k^{th} parse is a set $SV CQ_k = \{ SC^p \mid p \in P_Q^U \}$ where SC^p is a semantic chunk. Semantically viable chunk set. Formally we define semantically viable chunk sets as follows.

Definition. A *Semantically viable chunk set* (SVC set) of a query Q corresponding to the k^{th} parse is a set $SVC_{Qk} = \{ SC^p \mid p \in P_Q^U \}$ where SC^p is a semantic chunk. *Semantic chunk of a predicate p is defined as :*

- . If $CQ = \{ \}$, $SC^p = p$, $c1, c2, \dots, ci, \dots, cr (r \in \mathbb{N}^+)$, where ci or $ci = SC^p \cup SV CQ_k$, and ci is a condition on the predicate p

If $CQ = \{ \}$ (and $P^U = \{ \}$), $SC^p = p$

Such that, $SV CQ_k$ satisfies the following:

- a. $Q, SC^p \cup SV CQ_k$.
- b. $CQ, SC^p \cup SV CQ_k$ such that $SC^p = (p, c1, c2, \dots, ci, \dots, cr)$.

The condition ‘a’ states that there is a semantic chunk for each unknown predicates in the query. The condition ‘b’ states that each constraint in the query is used in at least one semantic chunk.

Definition. For a query Q , the semantically viable chunk set which is semantically valid as per the domain ontology is the *semantically valid chunk set*, $SV aCQ$. These sets are referred to as $SVaC$ sets.

We use syntactic information of the question to obtain the semantically viable chunk sets as discussed in the following section.

A. Finding Semantically Viable Chunk Sets

For a query, the main task for identification of semantically viable chunk sets is to identify the conditions for all the unknown predicates. We exploit syntactic information of the query for this purpose. We use a dependency-based parser (e.g. Stanford Parser,

Link Parser) to obtain the syntactic structure of the question. In case of syntactic ambiguity, these parsers provide all possible interpretations of the input sentence. In the following, we discuss the process of identifying the appropriate semantic chunks for different categories of queries.

Unknown Predicate as Noun. If an unknown predicate in the query plays the role of noun (e.g., ‘What is the role of Ritesh in AB Corp?’), its syntactic modifiers identify the constraints on the predicate. These modifier phrases give the constraints for the unknown predicate. The unknown predicate with its constraint is a candidate semantic chunk. For example, for the question ‘Give me the role of the associates with age > 30 years?’, the preposition phrase ‘with age > 30 years’ is a modifier of the noun ‘associates’. The constraint corresponding to this preposition phrase is ‘age > 30’, and hence the corresponding semantic chunk can be *name* obtained as $SC^{employee} = employee\ name, age > 30$.

Unknown Predicate as wh-word. In a domain ‘who’ usually refers to a person, such as ‘employee name’, ‘student name’; ‘when’ refers to date/time attributes like ‘joining date’, ‘completion time’; and ‘where’ refers to locations like ‘address’, ‘city’. For the given business application, this information about the wh-words is identified, and stored in the seed ontology. In questions involving any of these wh-word, the predicate corresponding to the wh-word is found using the domain ontology, which might be a possible candidate for being a unknown predicate. If the wh-word in the question is compatible to more than one predicate in the domain, then more semantic chunks -corresponding to each compatible predicate -are obtained. Semantic information is used in such cases to resolve the ambiguity regarding the most appropriate predicate (See Section V-B). The constraints of the wh-word are determined on the basis of the role of the whword in the question.

Using the syntactic information as discussed above, all possible semantic chunks for a parse structure of the question are determined. The set of these chunks is a semantically viable chunk set only if the chunk set satisfies the conditions (a) and (b) specified in the definition of SVC sets. For instance, for the query in Example 1, two SVC sets are obtained as:

$SV CQ_1 = \{ SC^{role}, SC^{employee\ name}, SC^{project\ name} \}$,
where:

$Q_1\ Q_1\ Q_1\ project\ name$

$= SC_{Q_1} = (project\ name, project\ type = SW\ ON, costing > \$15000, revenue < \$25000);$

– $SC^{employee\ name} = (employee\ name, joining\ date < 18/10/2008);$

$Q1$

– $SC^{role}_{Q1} = (role, SC^{project\ name}_{Q1}, SC^{employee\ name}_{Q1})$

And,

$SV\ CQ2 = \{SC^{role}_{Q2}, SC^{employee\ name}_{Q2}, SC^{project\ name}_{Q2}\}$, where:

– $SC^{project\ name}_{Q2} = (project\ name, project\ type = SW\ ON);$

$Q2$

– $SC^{employee\ name}_{Q2} = (employee\ name, joining\ date <$

$Q2$

$18/10/2008, costing > \$15000, revenue < \$25000);$

– $SC^{role}_{Q2} = (role, SC^{project\ name}_{Q2}, SC^{employee\ name}_{Q2})$

If for a query Q , only one semantically viable chunk set is found then this chunk set is the semantically valid chunk set.

In other cases, the semantically valid chunk set is found by using the domain specific semantic information as discussed in the following section.

B. Finding Semantically Valid Chunk Sets

If more than one semantically viable chunk sets are obtained for a question, semantic information obtained from the domain ontology is used to determine the semantically valid chunk set. Let $SV\ CQ_1 = \{SC^p, P_Q^U\}$ and $SV\ CQ_2 = \{SC^{Q1}, SC^{Q2}, P_Q^U\}$ be any two SVC sets for a query Q . Since there are more than one SVC set for Q , $pi, pj \in P_Q^U$, and $c = (p, v) \in Q\ CQ$ such that c is a constituent of SC^{pi} in $SV\ CQ_1$ and $Q1\ SC^{pj}$ in $SV\ CQ_2$. But, in the valid interpretation of Q , $c \in Q2$ can specify either the unknown predicate pi or the unknown predicate pj . Hence we conclude that, in this case, the syntactic information is not sufficient to resolve the ambiguity whether c is a constraint of pi or pj . For instance, consider the SVC sets of the query in Example 1 (Section V-A). In the two SVC sets obtained for this query, the constraints ‘costing > \$15000’ and ‘revenue < \$25000’ are bound to ‘project name’ in $SV\ CQ_1$, and to ‘employee name’ in $SV\ CQ_2$. To resolve such ambiguities, we use *depth* between the concerned predicates. The number of tables required to be traversed³ in order to find relationship between any two predicates is determined through the domain ontology. This is referred to as the depth between the

two predicates. If for a pair of predicates, there exists more than one path then we choose the one with the minimum depth. It is observed that the semantic chunk in which the unknown predicate and the constraint predicate pair has lesser depth is the one which is more likely to be the correct pair. We use domain ontology to find the depth between two predicates as described below.

Step 1. Breadth first search (BFS): The system does a BFS on the tables in the ontology to determine if pi or pj belongs in the same table as that of p . Without loss of generality, assume that pi and p belong to the same table, and pj does not belong to the table of p . In this case, SC^{pi} , $Q1$ and consequently $SV\ CQ1$ is assumed to be correct, and $SV\ CQ2$ is rejected. This in this case, $SV\ aCQ = SV\ CQ1$.

Step 2. Depth first search (DFS): We involve DFS method to resolve the ambiguity regarding the constraint c if BFS is not able to do so. The depth of the path from p to pi and pj is found using the domain ontology. The constraint c is attached to the predicate with which the distance of p is minimum, and the corresponding SVC set is the semantically viable chunk set. In Example 1 (Section III), the constraint predicates ‘revenue’ and ‘costing’ are found to be closer to the predicate ‘project name’ than to the predicate ‘employee name’. Hence, the semantically viable chunk set $SV\ CQ_1$ is the semantically valid chunk set. An advantage of this approach is that depending upon the question complexity the system does a deeper analysis. Domain ontology is used only if a question cannot be resolved by using just the syntactic information. If domain information also is not sufficient for question interpretation, then answers for all interpretations are found, and the user is asked to choose the correct answer.

VI. FORMAL QUERY FORMATION

The semantic chunks of the SVaC set are processed by the QueryManager. In this module, a formal query is generated on-the-fly from the semantic chunks to extract the answer of the user’s question. Since the domain ontology is in RDF format, we generate queries in SPARQL⁴ which is a query language for RDF.

For a semantically valid chunk set, we start with formulating SPARQL queries for the semantic chunks which do not contain any sub-chunk. The unknown

³This is found using the primary and foreign key information of the tables.

predicate of the semantic chunk forms the ‘SELECT’ clause, and the constraints form a part of the ‘WHERE’ clause.

For instance, from the SVaC set $SV\ aCQ$ (i.e., $SV\ CQ_1$ in Section V-A) of Example 1, we first formulate *project name* SPARQL query for the semantic chunks SC_{Q1} and *employee name* SC_{Q1} , and obtain the answer from the RDF. Assume that the answers of these chunks are obtained as “[Quantas, Bechtel, NYK]”, and “[Rajat, Nidhi]”. The answers obtained from independent semantic chunks are then substituted in the semantic chunks involving nested sub-chunks. For example, to formulate the query for the semantic chunk SC^{role} , the answers of the semantic chunks $Q1\ SC^{project\ name}$ and $SC^{employee\ name}$ are used. Therefore, $Q1\ Q1$ upon substituting these answers, the semantic chunk SC^{role} is modified as:

$SC^{role} = role, project\ name = Quantas, project\ name = Bechtel, project\ name = NYK, employee\ name = Rajat, employee\ name = Nidhi$. The SPARQL query for this chunk is then generated, and answer of the user query is retrieved.

VII. EXPERIMENTAL RESULTS

To test the approach discussed above, we carried out experiments on various domains (project management, retail, asset management). Users were asked to pose queries to an existing question answering system [4]. A set of almost 2750 questions were posed to the system, out of which approximately 35% consisted of fairly simple questions (No chunking required, but predicate-value binding required) e.g. “List all projects with costing less than \$30000”. The remaining 65% questions required correct predicate-value binding as well as correct chunking, like “list the employees in the projects having costing less than \$30000, started on 2009-10-10 with Ritesh as group leader”. When compared with actual answers following observations were made:

- 791 out of 946 questions were answered correctly for simple questions, which sums up approximately to 83.6%.
- For complex queries, 431 out of 1804 were correctly answered which approximately accounts 23.9%.

The users’ questions were again tested on the new system as discussed in this work. We have used Link Parser [5] for the syntactic analysis of the queries. The link parser yields the syntactic relations between different words of the sentence in the form of labeled

links. In case of ambiguity in the input sentence, the link parser yields syntactic structures corresponding to all possible interpretations.

The following observations were made:

- 863 out of 946 questions were answered correctly for simple questions, which sums up approximately to 91.2%.
- 1508 out of 1804 were correct from complex questions that accounts 83.6%

Comparing the results of the two approaches, a direct increase of about 7% was attained for simple questions and for complex queries, it went up by almost 58%.

VIII. CONCLUSION

We have described an approach to obtain domain specific chunk set for NL-enabled enterprise applications. For any question posed by a user to a business application system in natural language, the system should be robust enough to analyze, understand and comprehend the question and come up with the appropriate answer. This requires correct parsing, chunking, constraints formulation and sub-query generation. Although most general purpose parsers parse the query correctly, due to lack of domain knowledge, domain relevant chunks are not obtained. Therefore, in our work we have concentrated on enriching general purpose parsers with domain knowledge using domain ontology in the form of RDF. We have handled constraints formulation and sub query generation which form the backbone of any robust NL system. Tackling all these issues make any natural language-enabled business application system more robust, and enables it to handle even complex queries easily, efficiently and effectively.

REFERENCES

- [1] V. Lopez, E. Motta, V. Uren, and M. Sabou, “State of the art on semantic question answering -a literature review,” KMI, Tech. Rep., May 2007.
- [2] I. Androutsopoulos, G. Ritchie, and P. Thanisch, “Natural language interfaces to databases -an introduction,” *Natural Language Engineering*, vol. 1, no. 1, pp. 29–81, 1995.
- [3] G. Antoniou and F. van Harmelen, *A Semantic Web Primer*. The MIT Press, 2004.

⁴<http://dev.w3.org/cvsweb/2004/PythonLib-IH/Doc/sparqlDesc.html?rev=1.11>

- [4] S. Bhat, C. Anantaram, and H. K. Jain, "A framework for intelligent conversational email interface to business applications," in *ICCIT*, Korea, 2007.
- [5] D. Grinberg, J. Lafferty, and D. Sleator, "A robust parsing algorithm for link grammars," in *Fourth International Workshop on Parsing Technologies*, Prague, September 1995.
- [6] A.-M. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A. Yates, "Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability," in *20th international conference on Computational Linguistics*, Geneva, Switzerland, 2004.
- [7] B. Katz, G. Borchardt, and S. Felshinm, "Syntactic and semantic decomposition strategies for question answering from multiple resources," in *AAAI 2005 Workshop on Inference for Textual Question Answering*, Pittsburgh, PA, July 2005, pp. 35–41.
- [8] V. Lopez, E. Motta, and V. Uren, "Aqualog: an ontology-driven question answering system to interface the semantic web," in *NAACL on Human Language Technology*, New York, 2006, pp. 269 – 272.