A Novel Approach to Linked Data using Semantic Web and RDF

Gana Sindu K S[1], Ankitha Narayan[2] , Swetha Anandan[3], Monisha V. D[4]

ISE Dept. SaIT, Bangalore

Abstract—The Semantic Web is a technology for sharing data, just as the hypertext Web is for sharing documents. In this paper we briefly specify the semantic web and its uses. Further, by linking to other chosen resources, we could add immediate value to its page. While URIs are used for identifying things, the practice of supporting them with servers that would return relevant data, and links to Other data, has been confined to use of URIs for properties and classes and we made use of RDF for enabling the semantic web via linked data.

Keywords - RDF, Semantic Approach, URIs, Web, Server Data.

I. INTRODUCTION

We are now experiencing the bloom of the so called Web of Data. This term denotes the evolution of the web into an ecosystem of interconnected data an information contributed by the individuals. The word semantic web is defined as the web of data, which means all the data are interlinked together and it is a collection of all data possible. Semantic web is the extension of current web and the difference between them is that the information is given in well-defined meaning and collaboration of every bit of data so that it is better understood by people and computers.

Traditionally machines are used to store, organize, request, route, transmit, receive and display content encapsulated as documents. In order for machines to process the content of documents automatically –for whatever purpose-they primarily require two things: machine-readable structure and semantics. Unfortunately, despite various advancements in the area of Natural Language Processing (NLP) down through the decades, modern computers still struggle to link and manage 3-4 documents. The main problem is techniques to meaningfully process the structure and semantics of natural language due to ambiguities present in grammar. The current structure of the web's content is predominantly based around the idea of markup whereby the different elemental parts of the content in a document are delimited by the use of syntactic conventions, including matching start tags and end tags(e.g.,), nested elements, attributes and so forth. The eXtensible Markup Language (XML) provides a generic standard for markup-style languages, allowing machines to parse XML content into the data model consisting of an ordered tree of typed strings. Other non-markup-based methods for structuring content have also become common. For example, Comma Separated Values (CSV) provides a simple syntax that allows machines to parse content into tabular (or even relational) data-structures. Recently, JavaScript Object Notation (JSON) has seen growth in adoption, providing syntax to represent content that can be parsed into nested complex objects and associative arrays. On a high-level, the Semantic Web can be conceptualized as an extension of the current web so as to enable the creation, sharing and intelligent reuse of machine-readable content on the web. However, as far as machine is concerned, having formally structured content is only half the battle. Without some semantics (aka. meaning) for at least some parts of the content, machines would not be able to do much more than split the content up by its delimiter and load its structure.

The outlining core data model of semantic web defines it as the "Semantic Web Stack". The lower levels of the stack relate to foundational elements of the Semantic Web and for this, the Semantic Web relies on the standard Unicode character-set. Identifiers: if the Semantic Web is about describing things it can be conceptual or concrete in a machine-readable manner, these things will need globally agreed-upon identifiers. The natural choice for identifiers is thus to use the Uniform Resource Identifier (URI) specification, which is already used on the web to identify documents (or more accurately, representations). Syntax: to allow machines to automatically parse content into its elementary constituents, the Semantic Web requires syntaxes with formally defined grammar. For

this, existing generic syntaxes such as XML and JSON can be used. The core data-model elected for use on the Semantic Web is RDF, which can be serialized using one or more syntaxes. Thus, the Semantic Web requires formal languages with which to make claims about things described in RDF content. These formal languages offer a meta-vocabulary with well-defined semantics that can be used in combination with the RDF data-model. The real value of the web of data does not lie solely in the volume of data and information published online. Interestingly, it lies in the relationships between the data. These relationships put data in content and enrich their meaning and expressiveness.

II. LINKED DATA

Linked data is about connecting pieces of related dataand information coming fromdifferent sources e.g. information systems and databases.However,to make the Web of Data a reality, it is important to have the huge amount of data on the Web available in a standard format, reachable and manageable by Sematic Web tools. Furthermore, not only does the Semantic Web need access to data,but relationships among data should be made available,too,to create a Web of Data(as opposed to a sheer collection of data sets). This collection of interrelated data sets on the Web can also be reffered to as Linked Data.

To achieve and create Linked Data,technologies should be available for a common format(RDF),to make either conversation or on-the-fly access to existing databases(relational,XML,HTML etc). It is also important to be able setup query endpoints to access that data more conviently. W3C provides a palette of technologies(RDF,GRDDL,POWDER,RDFa,the upcoming R2RML,RIF,SPARQL)to get access to the data.Linked Data lies at the heart of what Semantic Web is all about:large scale integration of, and resoning on,data on the Web.

Examples : A typical case of a large Linked Data set is DBPedia,which,essentially,makes the content of Wikipedia available in RDF.The importance of DBPedia is not only that it includes Wikipedia data,but also that incoporates links to other datasets on the Web,e.g., to Geonames. By providing those extra links(in terms of RDF triples) applications can expoilt the extra(and possibly more precise) knowledge from other data sets whn developing an application; by virtue of integrating facts from several datasets,the application may provide a much better user experience.

Linked Data should:

1. Use Uniform Resource Identifiers(URIs) as names for things,e.g. http://dbpedia.org/resource/Brussels can be used for refering to the city of Brussels.

2.Use HTTP URIs,so that people can look up to those names.

3.When someone looks up a URI,provide useful information,using the standards(i.e RDF,SPARQL).

4.Include links to other URIs, so that more things can be discovered,e.g. from http://dbpedia.org/resource/Brussels a link is avaliable to http://dbpedia.org/resource /Belgium.

The above are the basic 4 rules to follow breaking them does not destroy anything,but misses an oppourtunity to make data interconnected.This in turn limits the ways it can later be reused in unexpected ways. It is unexpected re-use of information which is the value added by the Web.

The first rule,to identify things with URIs,is pretty much understood by most people doing semantic web technology.If it doesn't use the universal URI set of symbols,we don't call it Semantic Web.

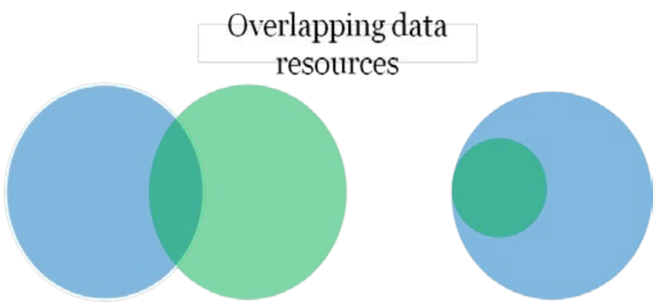
The second rule,to use HTTP URIs,is also widely understood.Sometimes it has to do with not understanding that HTTP URIs are names (not address)and that HTTP name lookup is a complex, powerful and evolving set of standards.

The third rule,that one should serve information on the web against a URI,in general,lookup the properties and classes one finds in data,and get information from the RDF,RDFS and OWL ontologies including the relationships between the terms in ontologies.

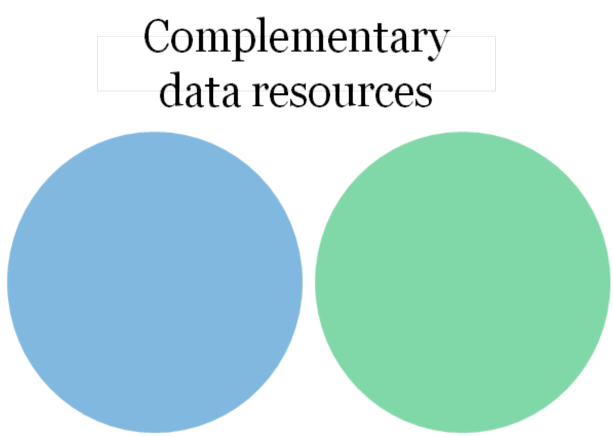
The fourth rule,to make links elsewhere,is necessary to connect the data we have into a web, a serious, unbounded web in which one can find all kinds of things, just as on the hypertext web we have managed to build.

There are many ways in which we can identify the relationship between the links

1.Overlapping data resources , i.e, data resources that refer to the same entity(often sharing some common information).



In this case,the linking takes place at unique indentifier level(i.e, URI) of the different data resources. For example,the DBpedia resource for the city of Brussels can be linked to one maintained by the Statistics Belgium and linking these two data resources allows us to get richer information about Brussels.



2. Complementary data resources, i.e. data resources that refer to different entities that somehow relate.

Imagine thatone of the attributes of the entity for the city Brussels is country.This attribute reveals that a city is positioned in /belongs to a country. In our case the value for countryis Belgium. There are different options for encoding this information.

One way would be to include the value for country as text,e.g. a literal or a string.This option however cannot take us too far and can suffer from different writings,different languages and even spelling errors.The Linked Data approach in this case opts for replacing the text value with a URI pointing to the specific country,i.e. to Belgium(the URI of DBpedia's resource forbelgium is http://dbpedia.org/resource/Belgium). The Linked Data option allows us to umambiguously refer to Belgium and also navigate through the links in order to collect more iformation about Brussels.

Basic web look-up

The simplest way to make Linked Data is to use,in one file,a URI which points into another.When you write an RDF file, sayhttp://example.org/smith ,then you can use local identifiers within the file,say #albert,#brian and #carol.In N3 you might say

<#albert> fam:child <#brian>,<#carol>. or in RDF/XML

<rdf:Description about="#albert"

<fam:child rdf:Resource="#brian">

<fam:child rdf:Resource="#carol">

</rdf:Description>

The WWW architecture now gives a global identifer http://example.org/smith#albert to Albert. This is a valuable thing to do,as anyone on the planet can now use that global idetifier to refer to Albert and give more information.

For example,in the document http://example.org/jones someone might write: <#denise> fam:child <#edwin>, <smith#carol> or in RDF/XML

<rdf:Description about="#denise"

<fam:child rdf:Resource="#edwin">

<fam:child rdf:Resource=http://example.org/smith/carol>

</rdf:Description>

Clearly it is reasonable for anyone who comes acrossthe identifier http://example.org/smith#carol to :

1. Form the URI of the document by truncating before the hash

2. Access the document to obtain information about #carol

We call this dereferencing the URI. This is basic Semantic Web

There are sevral varitions.

Variation:URIs without slashes and HTTP 303

There are circumstances in which dividing identifiers into documents doesn't work very well.There may be logically one global symbol per document,and there is reluctance to include

a # in the

URIsuchhttp://wordnet.example.net/antidisesablishmentairanism#word

Historically, the early Dublin Core and FOAF vocabularies did not have # in their URIs. In any event when HTTP URIs without hashes are used for abstract concepts, and there is a document that carries information about them, then :

1. An HTTP GET request on the URI ofb the concept returns 303 See Also and gives in the Location: header, the URI of the document.

2. The document is retrived as normal

This method has the advantage that URIs can be made up of all forms. It has the disadvantage that an HTTP request mBrowse-ableust be made for every single one. In the case of DublinCore, for example, dc:title and dc:creater etc are infact served by the same ontology document, but one does not know until the have each been fetched and returned HTTP redirections.

Varition:FOAF and rdfs: see Also

The Friend-Of-A-Friend convention uses a form of data link, but not using either of the two forms mentioned above.To refer to another person in a FOAF file, the convention was to give two properties, one pointing to the document they are described in, and the other for identifying them within the document.

<#i>foaf:knows [

foaf:mboxmailto:joe@example.com;

rdfs:seeAlsohttp://example.com/foaf/joe].

Read, "I know that which has email joe@example.com and about which more information is in http://example.com/foafjoe".

Infact, for privacy, often people don't put their email addresses on the web directly, but infact put a one-way hash(SHA-1) of their email address and give that.This clever trick allows people who know their email address already to work out that it is the same person, without giving the email away to others.

III. RDF

RDF stands for Resource Description Framework and is the main standard enabling the "Semantic Web" aka Linked Data or the "Web of Data". If HTML is the designed so that information and content is presented to human beings through a Web Browser, RDF is designed so that data is presented and possibly understood by machines. In essence, RDF is an Information Model, much like the Relational Database Model. In fact one could say that RDF is a Relational "Triple" Model. RDF is rooted in several forms of knowledge representation that have been researched in the field of Artificial Intelligence, such as the Entity Attribute Value (EAV), Semantic Networks and Frames. RDF and most of those models represent information as a collection of facts which tacitly forms a Graph or network of nodes connected with arcs. That is, RDF is very good for highly connected data e.g. the representation of social networks.

The basic building block or atom of information in RDF is called a Triple or Statement and represents a fact about a Resource.

The triple has the following form:

{Subject, Predicate, Object}.

Resources are represented by Resource Identifiers (IRIs, URIs, URLs) e.g. "http://myself.com" or Anonymous Identifiers e.g. "_:23452435". According to the standard Subject and Predicate fields can only be of type Resource. The Object field however can also be a Literal (plain or typed using the XML Schema Datatypes)

Example:

{http://me.com#Ale,http://myvocab.com#name, "Alejandro"},

{http://me.com#Ale, http://myvocab.com#age, "42" ^^http://http://www.w3.org/2001/XMLSchema…

Contrary to what most people think, RDF is not a data serialization method and it is not a flavour of XML (neither one of HTML). The confusion comes from the fact that RDF can be serialised using such mechanism, including XML and JSON. Now, considering you question I would say that RDF is absolutely the right approach to merge data from multiple sources. In fact most RDF store implementation extent the standard with a fourth field which contains an URI (IRI) denoting the Graph Name. The resulting Quad {Subject, Predicate, Object, and Graph} allows the construction of multi-graph data representations.

A web of linked RDF data may be enabled by standards specifying how links should be made in RDF and under what conditions the should be followed as well as powerful generic RDF browsers that can traverse an open web of RDF resources. The Tabular is an RDF browser, which is designed both for new users to provoke interest in the Semantic Web and give them a means to access and interact with the entire web of RDF data, and for developers of RDF content to provide incentive for them to post their data in RDF, to refine and promote RDF linking standards, and to let providers see how their data interacts with the rest of the Sematic Web. A challenge for Semantic Web browsers is to bring the power of domain-specific applications to a generic program when new unexpected domains can be encountered in real time. The Tabular project is an attempt to demonstrate and utilize the power of linked RDF data with a user-friendly Sematic Web browser that is able to recognize and follow RDF links to the other RDF resources based on user's exploration and analysis.

IV. CONCLUSION

The Semantic Web is the future of web and provides general views of arbitrary data that interlock seamlessly with specific meaning. The architecture of linked data proves to be a powerful one, and it is possible to build a generic data browser that provides sufficient functionality to make new data on the Semantic Web immediately viewable. That is, domain-specific applications will always be important and will always do better at specific tasks than the general one. This suggests that smooth interoperability between a generic client and an application-specific one. Perhaps the past lack of development of linked data is due to the fact that a harvester following links in general will attempt to load an unbounded set of data. And we described RDF which will give the possible framework for enabling the semantic web via linked data.

REFERENCES

[1] Herman.I.: The Semantic Web home page http://www.w3.org/2001/sw/(2006)

[2] Sander.C:BioPAX Level 2,Version 1.0.

[3] http://www.biopax.org/(2005)

[4] Bechhofer.S, van Harmelen, F.Hendler, J.Horrocks, I.McGuinness, D.L.Patel-Schneider, P.F.Stein, L.A: OWL Web Ontology Language Reference. Technical Report http://www.w3.org/TR/2004?REC-owl-ref-20040210/W3C(2004)

[5] Dean.M, Margerison.J. : PalmDAML

[6] http://www.daml.org/PalmDAML/(2002)

[7] Steer.D.: RDF Author.

[8] http://rdfweb.org/people/damian/RDFAuthor/(2003)

[9] Pietriga.E: IsaViz.

[10] http://www.w3.org/2001/11/IsaViz/(2006)

[11] Glaser.H, Alani.H, Carr.L, Chapman.S, Ciravegna.F, Dingli.A, Gibbins.N, Harris.S, M.C Schraefel, Shadbolt.N.In:CSAKTiveSpace: Building a Semantic Web Application. Springer Verlag(2004)417-432

[12] M.c.Schraefel, Karam.M, Zhao.S: mSpace: interaction design for user determined, adaptable domain exploration in hypermedia.In: AH 2003:Workshp on Adaptive Hypermedia and Adaptive Hypermedia and Adaptive Web Based Systems,Nottingham,UK(2003)217-235.