A Technique for Automatic Construction of Ontology from Existing Database to Facilitate Semantic Web

Debajyoti Mukhopadhyay, Aritra Banik, Sreemoyee Mukherjee

Web Intelligence & Distributed Computing Research Lab, Techno India Group
West Bengal University of Technology
EM 4/1, Salt Lake Sector V, Calcutta 700091, India
{debajyoti.mukhopadhyay, aritrabanik, m.sreemoyee}@gmail.com

Abstract

In this paper we describe a technique for automatic construction of ontology, a semantic representation of a conceptualization, for any domain, organization or enterprise whose Web site is existent with quiet a big mine of information. An user interactive system is developed with the help of which any organization, which has a Web site to maintain data, but has no corresponding semantic mark up, can automatically construct the ontology and further use that to model domain knowledge and make use of various Semantic Web applications like Semantic Web search engine etc. The automated construction of ontology will facilitate further the growth, popularity and usage of the Semantic Web and Semantic Web technologies as well as serve as a basis for Artificial Intelligence reasoning.

1.Introduction

Tim Berners-Lee, the inventor of the World Wide Web, defines the Semantic Web as "The Web of data with meaning in the sense that a computer program can learn enough about what the data means [in order] to process it" (Berners-Lee 1999) [12]. The Semantic Web is a set of technologies that envisions the existence of knowledge on the Web in a format that software applications can understand and reason about [3]. Thus, it would be possible to make knowledge more accessible to software and make software intelligent enough to understand knowledge and create new knowledge. Some elements of the Semantic Web are expressed as prospective future possibilities that have yet to be implemented or realized. Other elements of the Semantic Web are expressed in formal specifications. Some of these include Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL) [2][10]. All of which are intended to formally describe concepts, terms, and relationships within a given knowledge domain. It is thus a mesh of information linked up in such a way as to be easily processable by machines, on a global scale thus making the World Wide Web a universal medium for data, information, and knowledge exchange.

Thus, the Semantic Web is a framework that allows publishing, sharing, and reusing data and knowledge on the Web and across applications, enterprises, and community boundaries. Currently, the Semantic Web, consisting of the Semantic Web documents typically encoded in the languages RDF and OWL, runs parallel to the web of HTML documents. The Semantic Web essentially involves publishing data in a language called Resource Description Framework (RDF), specifically for data, so that it can be manipulated and combined just as can data files on a local computer [2]. RDF and the Web Ontology Language (OWL) which are ontology based procedures for representing knowledge on the web, providing useful features beyond those available in ordinary XML, allowing users to define terms (for example, classes and properties), express relationships among them, and specify constraints and axioms that hold for well-formed data. Representing knowledge using RDF first requires proper specification of that knowledge, if possible in a hierarchical manner, that is, describing the ontology. The Semantic Web is mostly built on syntaxes which use URIs or Uniform Resource Identifiers to represent data, which can be held in databases, reused and interchanged over the World Wide Web. One such set of syntaxes developed for achieving the above mentioned purposes is called the "Resource Description Framework" while another set of such syntaxes form the "Web Ontology Language" or OWL which is also used similarly to represent machine understandable knowledge. Therefore for any

0-7695-3068-0/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIT.2007.24

domain which wants to utilize Semantic Web technologies to improve performance and inculcate new features, requires corresponding semantic mark up which is nothing but the description of the domain ontology using RDF. Ontological representations based on subtype relationships provide the necessary knowledge base for AI techniques and representations for a variety of Semantic Web applications.

2. RDF

Before going into details of Resource Description Framework or RDF, it is essential to know the basis of RDF, that is, ontology. Ontology is nothing but a specification of a conceptualization [2]. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose [2]. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. It involves explicit representation of knowledge about some topic. Ontologies can enhance the functioning of the Web in many ways. They can be used in a simple fashion to improve the accuracy of Web searches—the search program can look for only those pages that refer to a precise concept instead of all the ones using ambiguous keywords. More advanced applications will use ontologies to relate the information on a page to the associated knowledge structures and inference rules. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data. Ontologies can be domain specific as well. Domain specific ontologies describe a specific domain by defining different terms applicable to that domain and specifying the interrelationship between them. Several domain ontologies can also be merged for a more general representation. Our work mainly involves development of a domain specific ontology described via one of the most common ontology language RDF. Development of ontology has several utilities. Firstly it helps in understanding an area of knowledge better, enables multiple machines to share the knowledge and use the knowledge in various applications. Resource Descripton Framework is nothing but a language to represent the ontology of an area of knowledge.

The Resource Description Framework (RDF) is a general purpose language for representing information on the web thus enabling exchange and reuse of structured metadata [9]. It provides a common framework for describing and interchanging metadata as well as well defined machine understandable semantics for metadata [5][9]. The XML syntax provides vendor independence, user extensibility, validation, human readability, and the ability to represent complex structures [10]. By exploiting the features of XML, RDF imposes structure that provides for the unambiguous expression of semantics and, as such, enables consistent encoding, exchange, and machine-processing of standardized metadata. In fact, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be *identified* on the Web, even when they cannot be directly *retrieved* on the Web. RDF is mainly intended for development of machine understandable information, that is, information that will be processed by applications rather than just for display. RDF comes with a common framework for expressing information so that information exchange can take place unhindered, without any loss of meaning. The ability to exchange information signifies that information can be made available to applications other than those for which it was created. RDF provides a model for describing resources called the RDF data model which consists of directed labeled graphs and model elements resource, properties, values. RDF defines a *resource* as any object described by RDF statements, each uniquely identifiable by a Uniform Resource Identifier (URI) and it may be an entire Web page or a part of a Web page [5][9]. On the other hand, "A property is a specific aspect, characteristic, attribute, or relation used to describe a resource" – W3C, Resource Description Framework (RDF) Model and Syntax Specification [4]. Note that a property is also a resource since it can have its own properties and all properties have specific values which may be atomic in nature (text strings, numbers, etc) or other resources which in turn may have their own properties.

An RDF statement combines a resource, a property and a value [4]. These three individual parts are known as the "subject," "predicate" and "object" [8]. For example, "Mr. Black works in Company X" is an RDF statement having the following parts:

Subject (Resource)	Mr. Black
Predicate (Property)	worksin
Object (value)	Company X

This can be represented as a directed labeled RDF graph as follows:

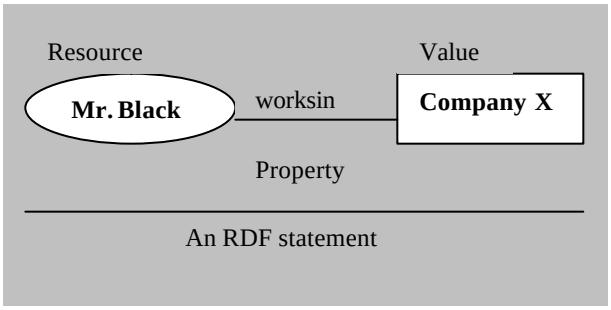

Fig. 1. An RDF directed labeled graph

The corresponding RDF code is as follows:

```
<?xml version="1.0"?>
<employee rdf:ID="Mr. Black"
xmlns:rdf="http://www.w3.org/1999
/02/22-rdf-syntax-ns#"
xmlns="http://www.westbengal.org/
crops#">
<worksin>Company X</worksin>
</employee>
```

This RDF code describes the resource Mr. Black which is an instance of the class Employee, has a property "worksin" whose value is "Company X" Separate RDF code depicts the class subclass relationships according as the RDF schema structure.

RDF Schema is a simple data-typing model for RDF [6] so that we can describe groups of related resources and the relationships among these resources. For example, we can say "Mr. Black" is an instance "Employee" and "Employee" is a subclass of "Organisation." The purpose of RDF schema is to express classes and their (subclass) relationships as well as to define properties and associate them with classes. The benefit of an RDF Schema is that it facilitates inferencing on the data, and enhanced searching. Resources can be divided into "classes" which are composed of instances. A class itself is also a resource which is usually identified by *RDF URI References* and can be described by *RDF properties*. We often use the prefix "rdfs:" to indicate the term is RDF Schema term. "rdfs:resource" is the root class of everything in RDF Schema. "rdf:type" is an instance of rdf:Property (class of RDF properties), and it means that a resource is an instance of a class. The property rdfs:subClassOf is an instance of rdf:Property that is used to state that a class is a Subclass of the other. The following RDF code shows that "Organisation" is the super-class and "Employee" is a subclass of "Organisation."

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/0
2/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/
01/rdf-schema#"
xml:base="http://
www.companies.org/organisation#">
<rdfs:Class rdf:ID="Employee">
  <rdfs:subClassOf
rdf:resource="#Organisation "/>
</rdfs:Class>
    ...
    ...
</rdf:RDF>
```

Another important RDF technique that we have utilized in our design methodology is the specification of the RDF domain and range.

Domain is used to indicate the classes that a property will be used with. One may specify zero, one, or multiple rdfs:domain properties.

Range is used to indicate the type of values that a property will contain. One may specify zero, one, or multiple rdfs:range properties [4][5].

The following RDF code snippet shows the usage of domain and range. It describes an RDF property "manager" whose domain is the Employee class i.e. it can be used with this class and its range is the "Department" class i.e. it can take values of the type "Department" only.

```
<rdf:Property
rdf:ID="seasonreqd">
        <rdfs:domain
rdf:resource="#Vegetable"/>
        <rdfs:range
rdf:resource="#season"/>
</rdf:Property>
```

Apart from the above mentioned basic features, RDF also provides means to express collection of items *rdf:bag, rdf:seq, rdf:alt, rdf:list rdf:rest* etc. Some other commonly used RDF syntaxes are *rdf:subject* (to state the subject of a statement), *rdf:predicate* (to state predicate of a statement), *rdf:object* (to state the object of a statement), *rdfs:isdefinedby* (to indicate a resource identifying the subject resource), *rdf:member* (to indicate a member of a subject resource), *rdf:comment* (a description of a subject resource) etc. RDF additionally provides means for publishing both human-readable and machine-processable vocabularies. Vocabularies are the set of properties, or metadata elements, defined by resource description communities. The standardized vocabulary declaration greatly encourages the reuse and extension of semantics across different information communities as well. Hence we can easily describe resources along with their properties, value of those properties, range, domain and also indicate interrelationship between the resources. Our work involves the automatic generation of this RDF coding from the existing data sources of any organization.

3. Our Approach for Ontology Construction

As has been mentioned before, for an increased usage of the Semantic Web technologies, it is essential that more and more domains are described using ontology languages like

RDF or OWL but most of the existing domains are developed using popular web development tools like HTML, jsp, servlet, asp or .net coding. Domains fully described using ontology languages are quite rare. This prompted us to devise a method and implementation of automatic ontology constructor software. A rather popular approach in this context is "conversion of Web-pages to RDF or OWL", the two most widely used efficient ontology representing languages.

Web-pages are a combination of some HTML, jsp, servlet or some asp or .net coding. RDF is a language which tells us "I know what it means, you tell me how it should look." On the other hand HTML is basically a schema which tells us how it (Web-page) looks with no relation to its meaning whatsoever. So it would be trying to convert a schema to a meaningful data source which is impossible in practical sense. Further more, if we assume it to contain several paragraphs of textual data, it would be a really time consuming work to understand the *proper* meaning from the data and extract sufficient information to construct an RDF page. We used a term 'proper meaning' which again can give rise to considerable confusion as meaning of any word varies according to the context of its usage. Some words have a meaning in one domain and something else in another domain. For example the word "bridge" carries different meanings in networking domain and in construction domain. Complete information about a domain or organisation is essential to develop automated ontology construction software for that domain.

We notice that the problem with Web-pages is that their data is not well defined so that a dumb like a computer can understand its meaning. Only in one place data is well defined that is a *database* or to be more specific a *relational database*. Here inter relation between data is considerably well defined. So we decided to concentrate on this and it resulted in an extraordinary finding.
If we consider a table as a class, it solves our problem.

EmpId	Name	Address	PhoneNo	DeptNo
E001	Xyz	Xyz	0000000000	D001
E002	Xyz	Xyz	0000000000	D002
E003	Xyz	Xyz	0000000000	D001

Fig. 2. Employee table

Say we are trying to build ontology for a company, some of whose database tables are Employee, Department, Inventory etc. designed for that organization. According to our development methodology, these relational database table names become the RDF class names of the RDF we intend to develop.

Now, instances of the *Employee class* should be the different employees working for that organization. Every relational database has a primary key. Primary keys help in the unique identification of data in a database. Say in the *Employee table* the primary key is *Employee_ID* attribute. Hence in the final RDF code to be generated from this table, each *Employee_ID* that stands for a unique employee becomes the instance. Other attributes such as name, address etc. become the properties of each instance generated. The RDF code corresponding to this table would be as follows:

```
<EMP rdf:ID="E001">
    <NAME> XYZ </ENAME>
    <Address> xyz </Address>
    <PhoneNo>0000000000</PhoneNo>
    <DeptNo> D001 </DeptNo>
</EMP>

<EMP rdf:ID="E002">
    <NAME> XYZ </ENAME>
    <Address> xyz </Address>
    <PhoneNo>0000000000 </PhoneNo>
    <DeptNo> D002 </DeptNo>
</EMP>

<EMP rdf:ID="E003">
     <NAME>XYZ</ENAME>
     <Address>xyz </Address>
     <PhoneNo> 0000000000 </PhoneNo>
     <DeptNo> D001 </DeptNo>
</EMP>
```

Now the question comes how to create properties from a database. Properties relate two classes. In this context we can say properties relate two tables. Now, it is known that, in RDBMS two tables are connected via foreign keys. In previous example *DeptNo* is the foreign key. So in terms of ontology we can say *DeptNo* is a property whose domain is Department and range is Employee. The corresponding RDF code would be as follows:

```
<rdf:Property rdf:ID ="DEPTNO">
  <rdfs:domain
rdf:resource="#DEPARTMENT"/>
  <rdfs:range rdf:resource="#EMPLOYEE"/>
</rdf:Property>
```

Now the *DeptNo property* of the *Employee class* becomes a pointer or url of the *Department class.* So the new structure would look like this:

```
<EMP rdf:ID="E001">
<NAME>XYZ</ENAME>
<Address>xyz</Address>
<PhoneNo>0000000000</PhoneNo>
<DeptNo>http://www.XYZcompany.org/
Department#d001</DeptNo>
</EMP>
```

Screenshots of automatic ontology generator and class structure creator are shown in Figure 3 and Figure 4.

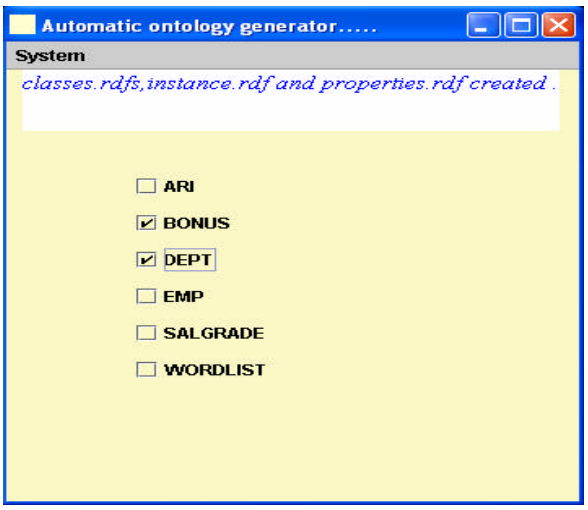

Fig. 3. Screenshot of the Automatic Ontology Generator

To find the class subclass interrelationships, a graphical user interface has been developed which prompts the user to enter the possible class subclass relationships and the software automatically generates the corresponding RDF codes. A screenshot of the class structure creator is as follows:

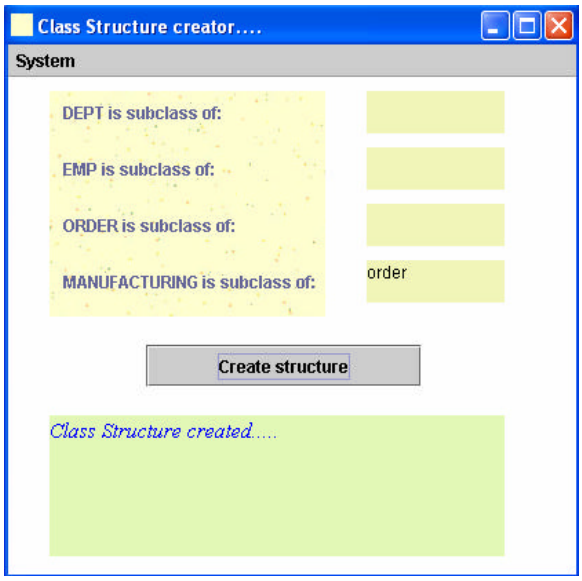

Fig. 4. Screenshot of the Class Structure Creator

The entire application has been developed using Java Swing. It gets connected to the database of the organization and produces corresponding RDF coding considering each and every table present in the database and their interrelationships.

4. Conclusion

The technique described above is entirely new and can be largely helpful in increasing the use of and popularizing of the Semantic Web techniques and taking up Semantic Web techniques would lead better exchange, reuse and storage of structured metadata. In the present WWW where most of the data is unstructured and HTML based, the use of Semantic Web applications is largely hindered. Populating the World Wide Web with sufficient metadata will pave the way for upcoming new and efficient technologies. First, searching on the web will become easier as search engines have more information available, and thus searching can be more focused. Useful technologies enabling automated software agents to roam the web can be developed, which would not only be searching for information but also would be transacting business on our behalf. The web of today, the vast unstructured mass of information, in the future can be transformed into something more manageable - and thus something far more useful. This brings up the requirement of efficient techniques to generate machine understandable information, that is, ontological representation of existing unstructured information. One such technique, developed by us utilizes the existing huge source of information in the form of relational databases and generates RDF coding for the data. This generated code can be used for multiple purposes, an example being Semantic Web search engines. Search engines which search for information based on the meaning of the entered data, can successfully utilize the generated semantic information in the form of RDF. Hence such a fast and efficient methodology for the automatic generation of ontology in the form of RDF from the existing Web-sites would lead to the generation of more and more machine interpretable information resulting in an unprecedented and effective growth of the upcoming and promising Semantic Web.

5. References

[1] F. Manola and E. Miller, eds. 2002; RDF Primer, W3C. http://www.w3.org/TR/rdf-primer.

[2] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*,5(2):199-220, 1993.

[3] T. Berners-Lee et al 2001. *The Semantic Web.* Scientific American. May 2001.

[4] W3C, *RDF Vocabulary Description Language 1.0: RDF Schema,* W3C Working Draft 23 January 2003, http://www.w3.org/TR/rdf-schema/.

[5] W3C, *Resource Description Framework (RDF) Model and Syntax Specification,* W3Crecommendation February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

[6] *An Introduction to the Resource Description Framework,* Eric Miller Research Scientist D-Lib Magazine May 1998.

[7] Sean B Palmer, The Semantic Web: An Introduction, 2001

[8] David E. Goldschmidt and Mukkai Krishnamoorthy; *Architecting a Search Engine for the Semantic Web.*

[9] Tim Bray, *What is RDF?* http://www.xml.com/lpt/a/2001/01/24/rdf.html.

[10] W3C, *RDF Primer,* W3C Working Draft 23 January 2003, http://www.w3.org/TR/2003/WD-rdf-primer-20030123/

[11] *Interview with Tim Berners-Lee,* Business Week, April 2007. The article, published in the same issue, also refers to numerous quotes from Tim.

[12] Berners-Lee.T,1999. *Weaving the Web:The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*,New York:Harper SanFrancisco.

[13] *OWL Web Ontology Language Guide*, W3C Recommendation, http:// www.w3.org/TR/oel-guide/.

[14] *A Guide to Creating Your First Ontology.* Natalya F. Noy and Deborah L. McGuinness.