

RDFS:An Extension To RDF

Gopal Pandey^{#1}, Prof. B.V.Buddhadev^{*2}

^{#1} Information Technology Department, Shantilal Shah Engineering College, Sidsar Bhavnagar, Gujarat, India

^{*2}Computer Engineering Department, L.D. College of Engineering Ahmedabad, Gujarat, India

Abstract— The data format to represent the Meta data of web resources is defined as the Resource Description Framework (RDF). RDF deficits in defining classes and their respective properties. While, RDF Schema (RDFS) is the mechanism of defining application-specific classes and their properties. Still, RDFS is just providing a framework to describe the classes and their properties. RDFS allows relating the classes as superclass-subclass. RDFS which incorporates class hierarchy can help in enriching the RDF. This paper discusses the limitations of the RDF and the basics of the RDFS using examples.

Keywords— RDF, RDFS, Semantic Web, XML.

I. INTRODUCTION

The RDF is defined to be a language that represents information about the resources on the WWW. It is more particularly intended for the representation of metadata of the web resources, which includes the title, author and the modification date of a webpage. In situations, where this information is needed to be processed by any application rather than just being displayed to the viewer, RDF is proposed. RDF ensures exchange of contextual information between the applications through a common framework without any loss of meaning. It represents a graph model besides using a well-documented XML schema for datatypes by identifying all elements by means of URIs. In RDF the structure of any expression is a triplet (Fig. 1), which consists of a subject, an object and a property.



Fig. 1. RDF representation using Graph.

To express simple statements about the resources, RDF helps out by providing the named properties and the values. However, the RDF user communities do need to have the abilities of defining the vocabulary (i.e. the terms) that they are intending to use in their own statements, particularly, suggesting that they are identifying specific form of resource or a class and will use specific properties for the description of these resources. RDF by itself does not provide any mechanism for defining application-specific classes and their properties. Rather, such classes and the properties are described to be a RDF vocabulary, using the extensions to RDF availed by the RDFS. The RDFS avails the facilities required for describing such classes and properties, and also to signify which combination of classes and properties to be used together. Otherwise, a type system for the RDF is rendered by the

RDFS. This type system of the RDFS is much similar to the type systems available in the object oriented programming languages such as the Java and the C++.

An RDFS property can also be considered to be an attribute of a class. These RFS properties can inherit resources from other properties where domain and range constraints can be of utilized to focus their purpose. For example, the domain constraint levies restriction of which property a class and classes may have and a range constraint restricts its possible values. With the help of these extensions, RDFS urges towards the current ontology languages. RDFS is helpful in declaring vocabularies for a particular community. As with RDF, to identify RDFS, the XML namespace mechanism is used. Any particular university can have classes defined as staff, AdminStaff and the AcademicStaff respectively. In Fig. 2, the superclass/subclass relationship is depicted graphically.

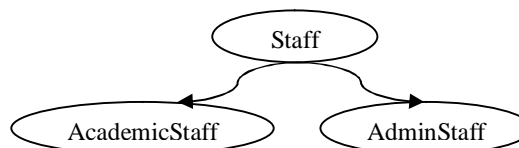


Fig. 2. A class hierarchy of staff

Fig. 3 illustrates the RDFS statements of Fig. 2, where classes and inheritance are used.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:sd="http://www.gtu.ac.in/staffdetails#">
  <rdf:Description rdf:ID="staff">
</rdf:Description>
  <rdf:Description rdf:ID="AcademicStaff">
<rdfs:subClassOf rdf:resource="#staff"/>
</rdf:Description>
  <rdf:Description rdf:ID="AdminStaff">
<rdfs:subClassOf rdf:resource="#staff"/>
</rdf:Description>
</rdf:RDF>
```

Fig. 3. RDFS examples of Class and subClass

II. RDFS ELEMENTS

A. RDFS Classes

Resources are the groups in which the RDF resources are divided. These resources can be known, using their URIs and described using their properties. Table I shows the different elements an RDFS class can have. Description of each element is mentioned below the Table I.

TABLE I. DIFFERENT ELEMENTS OF RDFS CLASSES

Element	Class of	SubClass of
rdfs:Class	All classes	--
rdfs:Datatype	Data types	Class
rdfs:Container	Containers	Resource
rdfs:Literal	Literal values (text and numbers)	Resource
rdfs:ContainerMembershipProperty	Container membership properties	Property

rdfs:Class is identical to the concept of a class in the object oriented programming languages. When a new class is defined by a schema, rdfs:type property with the value rdfs:Class is represented by that resource.

rdfs:Datatype is the class of data types and defines the allowed data types.

rdfs:Container is a collection of resources.

rdfs:Literal is the class of literal values such as strings and integers.

rdfs:ContainerMembershipProperty is a class that is used to state that a resource is a member of a container.

B. RDFS Properties

RDFS properties are the relation between subjects and objects in RDF triples. Table II shows the properties defined by the RDFS and are described below the given table. All properties have their domains and their ranges defined. "Domain" states that any resource is an instance of the class having the given properties. "Range" states that the values of a property are the instance of the class.

TABLE II. PROPERTIES OF RDFS ELEMENTS

Element	Domain	Range
rdfs:domain	Property	Class
rdfs:range	Property	Class
rdfs:subPropertyOf	Property	Property
rdfs:subClassOf	Class	Class
rdfs:comment	Resource	Literal
rdfs:label	Resource	Literal
rdfs:isDefinedBy	Resource	Resource
rdfs:seeAlso	Resource	Resource
rdfs:member	Resource	Resource

rdfs:domain suggests the classes on whose members, the properties can be used.

rdfs:range suggests the classes that the values of the properties must be member of.

rdfs:subPropertyOf is an instance of rdfs:Property specifying that on property is a specialization of another.

rdfs:subClassOf is a transitive property that suggests subset-superset relation between the classes.

rdfs:comment is a human-readable description of a resource.

rdfs:label is a string literal human-readable version of a resource name.

rdfs:isDefinedBy is a sub-property of rdfs:seeAlso and indicates the resource defining the subject resource.

rdfs:seeAlso specifies a resource that might provide additional information about the subject resource.

rdfs:member is a super-property of all the container membership properties.

III. RDFS EXAMPLE

Now, take into consideration the example of the members of the staff of any particular university. Class hierarchy of the academic staff is shown in Fig. 4. In this Figure, properties are depicted using the blocks and the classes are depicted using the ellipses. The superclass-subclass relationships between different classes are depicted in Fig. 4.

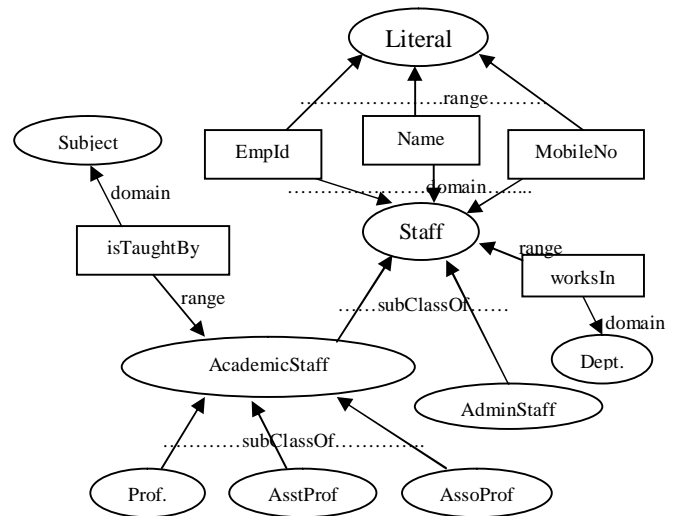


Fig. 4. A Class hierarchy of University Staff

With the help of the ontology of the university staff mentioned in the above Fig., RDFS statements can be described as shown in Fig. 5.

```

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
<rdfs:Class rdf:ID="Staff">
<rdfs:comment>
    The class of staff members of university
</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="EmpId">
<rdfs:comment>
It is an id of staff so property of staff members and takes
literals as values.
</rdfs:comment>
<rdfs:domain rdf:resource="#Staff"/>
<rdfs:range rdf:parseType="Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Name">
<rdfs:comment>
It is a name of staff so property of staff members and takes
literals as values.
</rdfs:comment>
<rdfs:domain rdf:resource="#Staff"/>
<rdfs:range rdf:parseType="Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MobileNo">
<rdfs:comment>
It is a mobile no of staff son property of staff members and
takes literals as values.
</rdfs:comment>
<rdfs:domain rdf:resource="#Staff"/>
<rdfs:range rdf:parseType="Literal"/>
</rdf:Property>

<rdfs:Class rdf:ID="AdminStaff">
<rdfs:comment>
    The class of Administrative staff of university
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#Staff"/>
</rdfs:Class>
<rdfs:Class rdf:ID="AcademicStaff">
<rdfs:comment>
    The class of academic staff of university
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#Staff"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Prof.">
<rdfs:comment>
    The class of Professor and all professor are academic
staff of university
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#AcademicStaff"/>
</rdfs:Class>
<rdfs:Class rdf:ID="AsstProf.">
<rdfs:comment>
    The class of Assistant Professor and all Asst.Prof. are
academic staff of university

```

```

</rdfs:comment>
<rdfs:subClassOf rdf:resource="#AcademicStaff"/>
</rdfs:Class>
<rdfs:Class rdf:ID="AssoProf.">
<rdfs:comment>
    The class of Associate Professor and all Asso.Prof. are
academic staff of university
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#AcademicStaff"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Subject">
<rdfs:comment>The class of subjects offered by university
</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="isTaughtBy">
<rdfs:comment>
Inherits its domain from "Subject" Class and range from
"AcademicStaff" Class
</rdfs:comment>
<rdfs:domain rdf:resource="#Subject"/>
<rdfs:range rdf:resource="#AcademicStaff"/>
</rdf:Property>

<rdf:Property rdf:ID="worksIn">
<rdfs:comment>
Inherits its domain from "Dept" Class and range from "Staff"
Class
</rdfs:comment>
<rdfs:domain rdf:resource="#Dept"/>
<rdfs:range rdf:resource="#Staff"/>
</rdf:Property>

</rdf:RDF>

```

Fig. 5. RDFS Example of Fig. 4

IV. LIMITATION OF RDFS

RDFS allows us to express semantic knowledge in terms of object model. But RDFS is missing some features as mentioned below.

- Local scope of property: RDFS cannot express equivalence between properties. This is important to be able to express the equivalence of ontological concepts developed by separate working groups. For example, "fatherOf" property indicates the same class like elephants are the father of elephants while mice are the father of mice
- Cardinality restriction: RDFS does not have the capability of expressing the uniqueness and the cardinality of properties. In some cases, it may be necessary to express that a particular property value may have only one value in a particular class instance. If we want to place the restriction on how

many distinct values a property may take. For example, we would like to say that a subject is taught by at least one lecturer.

- Disjointness of classes: RDFS cannot express disjointness. For example, the gender of a person can be male or female. While it is possible in RDFS to express that Ramesh is a male and Radha a female, there is no way of saying that Ramesh is not a female and Radha is not a male. Here male and female classes are disjoint class.
- Boolean combination of Classes: RDFS cannot express the concept of unions and intersections of classes. This allows the creation of new classes that are composed of other classes. For example, the class "Staff" is the disjoint union of the classes like "AdminStaff" and "AcademicStaff" refer Fig. 4.
- Special characteristics of properties: RDFS can express the values of a particular property but cannot express that this is a closed set. For example, an enumeration for the values for the gender of a person should have only two values: male and female. Property may be transitive like "lessThan", unique like "hasMother" or inverse of another property like "employee" and "employer".

CONCLUSION

To conclude, RDF schema (RDFS) does allows users to define resources with classes, properties and values. The concept of RDF classes is very much related to the OOP's based programming languages such as Java and C++. Where class can be inherited and is composed of similar entities. This further allows resources to be defined as instances of these classes and derived class of the same classes. For example, resources can be defined as the instances of one or more classes in RDFS where these classes can also be arranged in hierarchical manner.

In an RDF model the classes of resources and properties used is described by the RDFS vocabularies. With the help of RDFS, classes and properties can be arranged in

specialization/generalization hierarchies, domain and range expectations for properties can be defined, class memberships can be maintained, and data types can be specified besides interpreting them. All resources in RDFS are considered members of the class of all RDF resources and as such are all instances. These instances can be further depicted by making statements of the properties or by making them a member of other classes found in the RDFS vocabulary explicitly.

REFERENCES

- [1] Fensel, Hendler, Lilberman & Wahlster, "Spinning the semantic web" editors - Publisher : MIT Press.
- [2] Gopal Pandey, "Semantic Web: An Introduction and Issues" , International Journal of Engineering Research and Applications(IJERA), Vol.2 issue 1, jan-feb 2012 page 780-786.
- [3] O. Lassila and R. Swick., "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation, 1999. Internet: <http://www.w3.org/TR/REC-rdf-syntax/>.
- [4] G. Klyne and J. Carroll., "Resource Description Framework (RDF): Concepts and Abstract Data Model", W3C Recommendation, 2004. Internet: <http://www.w3.org/TR/rdf-concepts>.
- [5] Patrick Hayes, "Rdf semantics" W3C Recommendation, 2004. Internet: <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>.
- [6] D. Beckett, "RDF/XML Syntax Specification (Revised)". W3C Recommendation, 2004. Internet: <http://www.w3.org/TR/rdf-syntax/>.
- [7] Dean Allemang, Jim Hendler, "Semantic Web for the Working Ontologist", MK Publisher.
- [8] Frank Manola, Eric Miller, "RDF Primer", W3C Recommendation, 2004. Internet : <http://www.w3.org/TR/2004/REC-rdf-primer-20040210>.
- [9] Jan Grant, Dave Beckett, "RDF Test Cases", W3C Recommendation, 2004. Internet: <http://www.w3.org/TR/2004/REC-rdf-testcases-20040210>.
- [10] D. Brickley and R.V. Guha, " RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation. February 10, 2004. Internet: <http://www.w3.org/TR/rdf-schema>.
- [11] Jorge cardoso, "Semantic Web Service Theory, Tools and Applications" Information Science Reference.
- [12] Grigoris Antoniou, Frank van Harmelen, "A Semantic Web Primer", Publisher : MIT Press.
- [13] Toby Segaran, Colin Evans, Jamie Taylor, "Programming the Semantic Web", Publisher: O'Reilly.
- [14] Shelley Power, "Practical RDF" , Publisher : O'Reilly.