Online Music Library using RDF based Inference Engines

Kumar Abhishek,
Dept.of CSE
NIT Patna-80005
India

Gopinath.V
Dept. of CSE ,
MIT Manipal-576104
India

Vipin Kumar.N
Eucation and Research, Mangalore SEZ , Infosys technologies, India

Archana P. Kumar
Dept. of CSE,
MIT Manipal-576104,
India

ABSTRACT

The paper describes the method of how to develop “Intelligent agents” for web searching and replacing the existing “User agents” with the “Intelligent agents”. This process is described by developing a website (viz.Music Library) which runs on an inferencing engine which suggests the users the kind of music they like based upon its reasoning. The database for such an engine has been moved away from the traditional RDBMS format and more easy-to-use XML files have been used in place. These XML files are in fact RDF/XML files, which allow inferencing process to be achieved in a smooth fashion.

Keywords

Intelligent agents, Inferencing, RDF/XML format.

1. INTRODUCTION

There has been a constant effort from the W3C community to develop the World Wide Web in a direction that makes surfing as simple, efficient and intelligent as possible.”Simple” by moving most of the work loads from the users to the machines “Efficient”, in a sense making search results more meaningful and machine understandable “Intelligent”, in the sense of making machines work and understand the content of the web without any human intervention. This process also involves machines interacting with each other all by themselves and giving meaningful results. This process would bring around some level of artificial intelligence in computers.

1.1 Intelligent Agents

Intelligent agents are “user” like agents with “intelligence” embedded into them. Making intelligent means making them understand the web of data as understood by humans.

2. RESOURCE DESCRIPTION FRAMEWORK

Resource Description Framework was originally developed as a metadata model, the first specification of RDF came in the year 1999 authored by Ora Lassila and Ralph Swick [2].

RDF is based on XML and is defined as a language used for representing information about resources in the World Wide Web that is adding metadata to the web resource [2].

The basic idea of RDF is to identify things (resource) using Web Identifiers. This web identifiers are known as Uniform Resource Indicator (URI). RDF also describes resources in terms of properties and properties value [2].

RDF is also known as TRIPLES where triple is a magic number that is three piece of information needed to fully define a single bit of knowledge. The three pieces of information are subject, property type and property value. For example I (subject) have a name (property), which is Kumar Abhishek (property value) [3].

In English grammar rule a complete sentence (or statement) contains two things a *subject* and a *predicate*: the subject is the who or what of the sentence and the predicate provides information about the subject [3]. For example

The title of the article is "*Pranab Roy*."

In the above example subject is the article, and the predicate is title, with a matching value of "Pranab Roy" [3].

If the above English statement is translated to an RDF triple, the *subject* is the thing being described—in RDF terms, a *resource* identified by a URI and the *predicate* is a property type of the resource, such as an attribute, a relationship, or a characteristic [3].

Apart from subject and predicate the specification also introduces a third component, the *object* [3].

In RDF, the object is equivalent to the value of the resource property type for the specific subject [3].

RDF core committee decided to represent the data model in RDF using directed label graph [3].

The RDF directed graph consists of a set of nodes connected by arcs, forming a pattern of *node-arc-node*. The nodes come in three varieties: *uriref*, blank nodes, and literals [3].

The uriref consists of a Uniform Resource Identifier (URI) reference that provides a specific identifier unique to the node. Blank nodes are nodes that don't have a URI. The literals consist of three parts—a character string and an optional language tag and data type. Literal values represent RDF objects only, never subjects or predicates. In RDF literals are represented by drawing rectangles around them [3].

2.1 RDF Data Model

The representation of data in RDF is done via graph. In RDF as we know it comprises of three things subject-object and predicate, in RDF statement the term subject is either a Uniform Resource Identifier (URI) or a blank node, which denote resources [3].

Resources which are represented by blank nodes are termed as anonymous resources. These resources are not directly identified from RDF statement.

The term predicate is also an URI which indicates a resource, representing a relationship.

The object is a URI, blank node or a Unicode string literal [3].

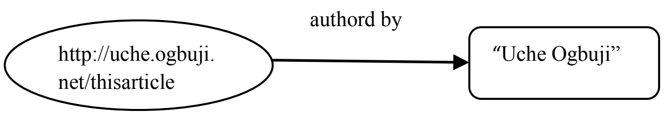

Figure 1- RDF Statement [7]

Figure 1 depicts a graph representation of RDF statements [7].

In the figure above, the object is a string: "Uche Ogbuji". Then the object is termed as *literal* in RDF, but an object could also be a resource [7].

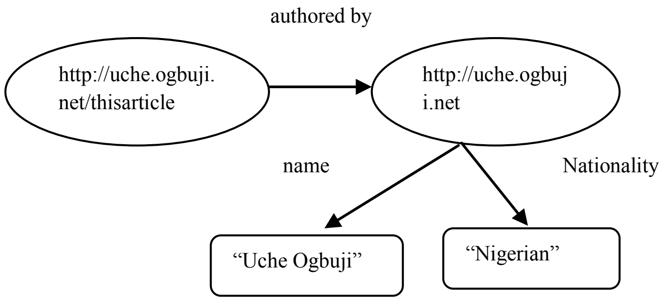

Figure 2: A small RDF model [5]

Figure 2 depicts combination of several RDF statement into a single diagram. The expansion of RDF is done on this basis.

RDF describes a Web-based resource by defining a directed graph of statements.

In the above figure, the object which is a literal "Uche Ogbuji" is replaced by a URI indicating this person, which in turn is the subject of several more statements. This type of collection of RDF statements is termed as *model* in RDF [7].

Uniform Resource Locator are used to point to Web Documents that describe the exact meaning (semantics) of each edge type.

3. INFERENCE ENGINE

Inference engine or a semantic reasoner is a software tool which is used to derive new knowledge from already existing knowledge by using rules, called as inference rules. The inference rules control all the steps for inference which is developed by the inference engine [8].

Semantic web makes use of inference engine to process the knowledge available. Let us take for instance

Grandfather (Tony, Mac) |: Father (Tony, John) &
Father (John, Mac)
There are two related to the above consideration:

(a) Tony is father of John
(b) John is father of Mac

The rule devised here is ---"when we find a new relationship where Tony is a father of John and then for second consideration, that John there is a Mac for which Jack is the father of Mac, then new knowledge is Tony is a grandfather of Mac [8].
Ontology language is used to specify inference rule. While writing inference many semantic reasoner makes use of first-order predicates and based on this there two basic types of inferencing

i. Backward Chaining
ii. Forward Chaining

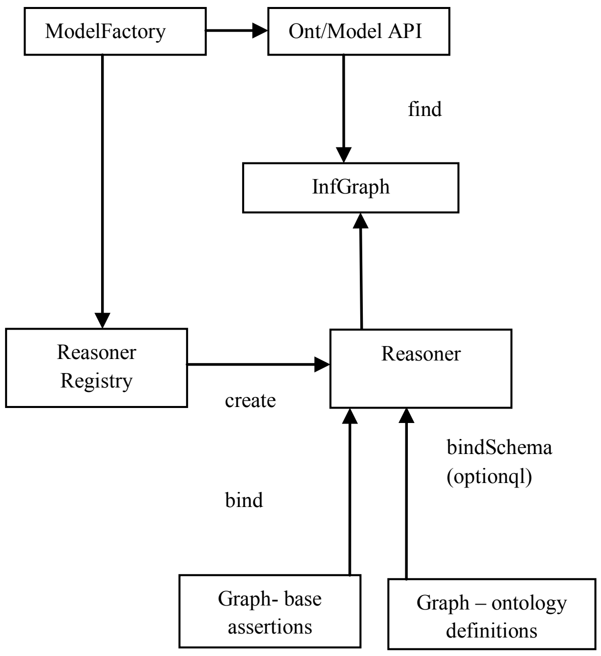

Figure 3 – Inference Machinery [9]

The Backward Chaining method tries to achieve the goal by working backward that is it tries to prove a goal by finding out the truth of its condition [8].

Let us take an example of rule "if A and B then C", the backward inference will prove C by first proving C and then proving B [8].

The Forward Chaining method is also known as data-directed inference, i.e. data gets itself in working memory. When the data is put in the working memory this will trigger rules whose condition should match new data. The rules will perform actions which may result in adding new data in memory which will trigger more [10].

Many of the inference engine makes use of forward chaining, some of the most common inference engine used for semantic web application are – F-OWL, JENA, RACER etc [8].

4. METHODOLOGY

The website constructed is Music Library portal. Unlike the contemporary libraries where the user just clicks on his type of music and the songs related to that genre, this portal is more of a suggesting nature. Based on the genre of music the user likes it gives its opinion on what other type of music the user might be interested in.The following tools are used in the process of creation of the portal:

Protégé: A tool developed by Stanford University for creation of RDF/XML files in a simple fashion.

*Jena :*A Java API for creating models for the inferencing process.The API also has methods to access the RDF files,for accessing the inference engine and putting out the results in a required format(here HTML format).

*Eclipse :*An IDE for the writing the required code for the Jena applications,and methods of how the user end and the database end connect.

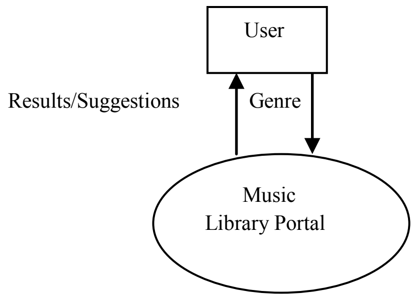

Figure 4- User selection

The user in Figure 4 is allowed to choose a genre of his choice.The program running at the server displays the result and also gives a suggestion based on the genre.

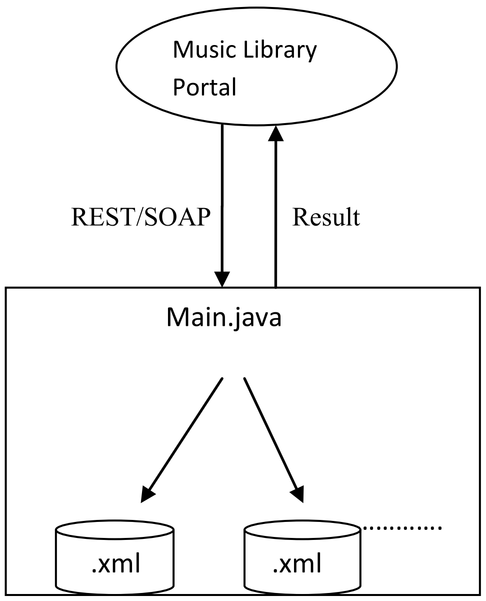

Figure 5- Backend database access

The browser at the client end accesses the web server. The server has a program running namely Main.java .This file has the capability of accessing the repository at the backend. The backend here is a database of XML files.

Important aspects of Main.java file:

- Creating a Jena model
- Copying the required RDF file into the model (The Jena creates a graph structure of the rdf data).
- Initializing the reasoner using the imported packages.
- Switching/Calling to/the required query.
- Putting the result of the query into the required format.
- Sending the result back to the browser to display.

4.1 JENA Model

This step is used for creating a default model to be used by Jena.This model will be used for the processing of the RDF files.

```
import com.hp.hpl.jena.rdf.model.Model;        Model
model=ModelFactory.createDefaultModel();
```

4.2 Copying RDF file

```
InputStream in=null;
in=new FileInputStream(new File("<filename>");
model.read(in,null);
```

This snippet is used to copy the required RDF file from the database into the empty Jena model created, for the processing of queries.

4.3 Initializing the Reasoner

```
Reasoner
reasoned=RDFSRuleReasonerFactory.theInstance().create(nul
l);

 InfModel inf =ModelFactory.createInfModel(reasoner,

rdfsExample);
```

This snippet initializes the RDF reasoned present in the Jena API and also initializes an Inference model.This Inference model is used to hold results generated by the reasoner.

External reasoners can be accessed through HTTP by using the following snippet:

```
ReasonerManager reasonerManager =
ReasonerManager.getInstance();

        ProtegeOWLReasoner reasoner =
reasonerManager.getReasoner(model);

 if (reasoner.isConnected()) {

   DIGReasonerIdentity reasonerIdentity =
reasoner.getIdentity();

   System.out.println("Connected to " +
reasonerIdentity.getName());

}
```

4.4 Calling the query

```
String a=null;

a="PREFIX
lib:<http://www.music.com/ontologies/music.owl#>"+

"PREFIX foaf:<http://www.w3.org/2000/01/rdf-schema#>"+

            "SELECT ?a "+

            "WHERE {"+

" ?a foaf:subClassOf lib:JazzAndMetal" +

" }";
```

Calling the required query(here the subclasses of class Jazz And Metal).

4.5 Output and format setter

Query query = QueryFactory.create(a); QueryExecution qe = QueryExecutionFactory.create(a, model); ResultSet results = qe.execSelect(); ResultSetFormatter.out(System.out, results, query); OutputStream o=new FileOutputStream("C:/Protege/c.html"); byte b[]; b=ResultSetFormatter.asXMLString(results). getBytes();

The query execution class has methods which can execute the given SELECT query. The method execSelect() does this.The rest of the snippet is used to format the result as preferred by the user.

5. EXPERIMENTAL RESULTS

The RDF file had no subclasses for the Jazz and Metal class. It is up to the reasoner to classify the class based on the type of instrument and to the interest of the user. Intially if the user chooses a genre say, Black metal, the reasoner uses its rules to classify and reasons out that if a user likes black metal then a user might like the instruments "Drums" and if a user likes drums then he might like another genre which has heavy usage of drums in it(in this case its Cool Jazz).So instead of just giving Black Metal as the result the server is going to suggest the user with Cool Jazz.In this way the machine can interpret data based on First Order Logics and without any human intervention.

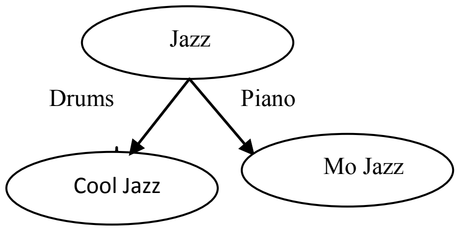

Figure 6 – Classification of Jazz genre

The above figure describes two genres of the Jazz class. The class Cool Jazz has "Drums" as its vital instruments and Mo Jazz as "Piano".

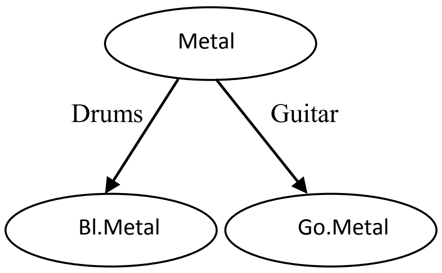

Figure 7 – Classification of metal genre

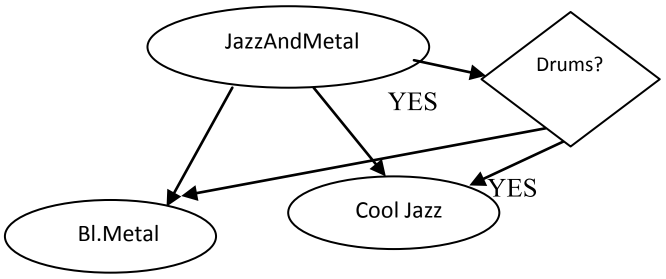

Figure 8 – Classification by reasoner

6. CONCLUSION

The RDF model described above can be helpful in creating machines which can communicate with each other and take decisions on their own. The traditional RDBMS methods are a bit structured and difficult to maintain. The RDF models can give solutions to such problems as they are semi-structured and flexible. Since RDF models have the XML structure, a level of intelligence, ease if coding that intelligence into machines and maintaining these files is much easier than the traditional tables of relational database.

7. FUTURE WORK

It is fast and efficient results which are the key to any search engine's success. Implementation of such reasoners into RDF models can be the stepping stone to the world of AI.

Work involving better storage mechanisms for XML files and for providing secure mechanisms for extracting and delivering them.

Improvement in the field of rule based inference engines can also help in movement of the Web from 2.0 into 3.0.

8. REFERENCES

[1] W3C Semantic Web Frequently Asked Questions". W3C. http://www.w3.org/2001/sw/SW-FAQ. Retrieved March 13, 2008.

[2] W3C http://www.w3.org/TR/rdf-primer

[3] Shelley Powers: Pratical RDF, O'Reilly,2003

[4] Michael Grobe : RDF, Jena, SparQL and the "Semantic Web"

[5] Eclipse: http://www.eclipse.org/downloads/

[6] Jena and ARQ(a SPARQL implementation) are representative for various API's for RDF processing.

[7] IBMhttp://www.ibm.com/developerworks/library/w-rdf

[8] Gizem Olgo "Inference Engines-Semantic Web Cross Up Project", July 2004

[9] Jena 2 Inference support, available at http://jena.sourceforge.net/inference/

[10] Jocelyn Ireson –Paine "http://www.jpaine.org/students/lectures/lect3/node10.html"