RESEARCH PAPER | Computer Science | Volume : 4 | Issue : 1 | Jan 2014 | ISSN - 2249-555X

A Study on Semantic Web Framework: JENA and Protégé

KEYWORDS: Semantic web ,RDF Graph, Ontologies

Kruti Jani
Assitant Professor, SCCPGICA, Ahmedabad

Dr. V.M. Chavda
Principal, NPCCM, KADI

ABSTRACT *The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Jena is an open source Semantic Web framework for Java for developing Semantic Web applications. It can be used to create and manipulate RDF graphs. Protégé is a free open source Ontology Editor and Knowledge Based Framework. It supports modeling ontologies. Both can be used for similar tasks with the some differences. This paper discusses comparative study of two frame work Jena and Protégé. Frameworks like this are worth investing in, since they might play the key role in the evolution of the WWW into Semantic Web.*

I. INTRODUCTION

The term "Semantic Web" is often used more specifically to refer to the formats and technologies that enable it [3]. These technologies include the 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), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain. RDF graph can be modeled by using the Java based framework called Jena. Jena based on Java deals with programmatic statements. The same can be done by using an editor- Protégé. The paper compares how the semantic web concepts can be designed and modeled using the two API and also states as to which API should be used while developing the Semantic Based Web Applications for better performance metrics.

A) JENA FRAMEWORK

Jena is a Java framework for building Semantic Web applications [1]. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Jena is open source and grown out of work with the HP Labs Semantic Web Program [5]. The Jena Framework includes:

- A RDF API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- An OWL API
- In-memory and persistent storage
- RDQL- a query language for RDF
- SPARQL query engine

Jena is a Java API which can be used to create and manipulate RDF graphs. Jena has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively[2]. A graph is called a model and is represented by the Model Interface [4]. To build applications exploiting the ontology, we need an API allowing us to access and manipulate directly an ontology written in OWL Since it is reliable, mature and offers a good compatibility with most of the other RDFS/OWL API.

Thus Jena Classes can be used to:

- Retrieve and Parse an RDF File containing a graph or a collection of graphs.
- Store it in memory
- Examine each triple in turn, examine one component(say, the subject) of each triple in turn, or examine only triples that meet specified criteria and
- Write a serialized version of a graph to a file.

An RDF Graph is stored in Jena as a "model", and a Jena model is created by a factory, as in:

Model m=ModelFactory.createDefaultModel();

Once a model has been defined, Jena can populate it by reading data from files, backend databases, etc. in various formats and once it has been populated, Jena can perform set operations on pairs of populated models and /or search models for specific values or combinations (patterns) of values.

B) RESOURCE DESCRIPTION FRAMEWORK (RDF)

The Resource Description Framework (RDF) is a standard (technically a W3C Recommendation) for describing resources [5]. A Resource is anything we can identify Every Resource has a URI, a Universal Resource Identifier[2]. A URI can be a URL or some other kind of unique identifier. RDF is best thought of in the form of node and arc diagrams. Each arc in an RDF Model is called a statement. Each parts:

- the subject is the resource from which the arc leaves
- the predicate is the property that labels the arc
- the object is the resource or literal pointed to by the arc

A statement is sometimes called a triple, because of its three parts.

A simple vcard might look like this in RDF:

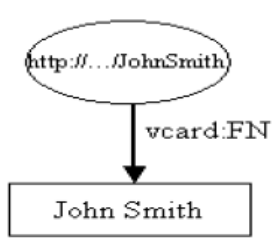

Fig -1 simple vcard in rdf with more details of the person node[5]

The resource, John Smith, is shown as an elipse and is identified by a Uniform Resource Identifier (URI) [1], in this case "http://.../JohnSmith". Resources have properties. In these examples the sort of properties that would appear on John Smith's business card. Figure 1 shows only one property, John Smith's full name. Properties are usually represented in this qname form when written as RDF XML and it is a convenient shorthand for representing them in diagrams and in text.

Volume : 4 | Issue : 1 | Jan 2014 | ISSN - 2249-555X

Strictly, however, properties are identified by a URI. Each property has a value. In this case the value is a literal, which for now we can think of as a strings of characters. Literals are shown in rectangles. Jena is a Java API which can be used to create and manipulate RDF graphs like this one. Jena has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively. In Jena, a graph is called a model and is represented by the Model interface.

The code to create this graph, or model, is simple:

```
// some definitions

static String personURI = "http://somewhere/JohnSmith";

static String fullName = "John Smith";

// create an empty Model

Model model = ModelFactory.createDefaultModel();

// create the resource

Resource johnSmith = model.createResource(personURI);

// add the property

johnSmith.addProperty(VCARD.FN, fullName);
```

The John Smith resource is then created and a property added to it. The property is provided by a "constant" class VCARD which holds objects representing all the definitions in the VCARD schema. Jena provides constant classes for other well known schemas, such as RDF and RDF schema themselves, Dublin Core and DAML.

The code to create the resource and add the property, can be more compactly written in a cascading style:

```
Resource johnSmith = model.createResource(personURI)
.addProperty(VCARD.FN, fullName);
```

Another example of representing different parts of John Smith's name can be:

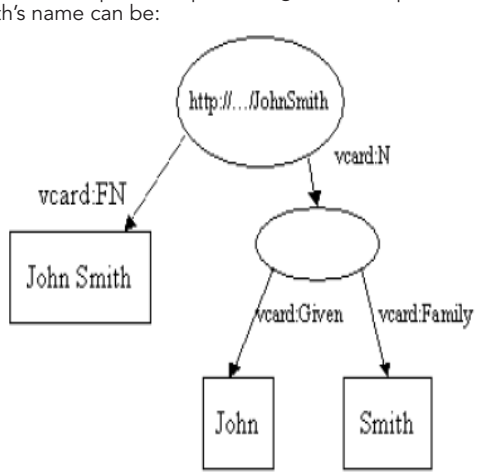

Fig. 2: simple vcard in rdf with more details of the person node[5]

The Jena code to construct this example, is again very simple. First some declarations and the creation of the empty model.

```
// some definitions

String personURI = "http://somewhere/JohnSmith";

String givenName = "John";

String familyName = "Smith";

String fullName = givenName + " " + familyName;

// create an empty Model

Model model = ModelFactory.createDefaultModel();

// create the resource

// and add the properties cascading style

Resource johnSmith

= model.createResource(personURI)

.addProperty(VCARD.FN, fullName)

.addProperty(VCARD.N,

model.createResource()

.addProperty(VCARD.Given, givenName)

.addProperty(VCARD.Family, familyName));
```

Jena has methods for reading and writing RDF as XML. These can be used to save an RDF model to a file and later read it back in again. The key RDF package for the application developer is com.hp.hpl.jena.rdf.model. The API has been defined in terms of interfaces so that application code can work with different implementations without change. This package contains interfaces for representing models, resources, properties, literals, statements and all the other key concepts of RDF, and a ModelFactory for creating models. So that application code remains independent of the implementation, it is best if it uses interfaces wherever possible, not specific class implementations.

II. PROTEGE

Protégé is a free open source Ontology Editor and Knowledge Based Framework developed and published by Stanford University[6]. It is published under the terms of Mozilla Public License[7]. Protégé is a flexible, configurable platform for the development of arbitrary model-driven applications and components. Protégé has an open architecture that allows programmers to integrate plug-ins, which can appear as separate tabs, specific user interface components (widgets), or perform any other task on the current model. The Protégé-OWL editor provides many editing and browsing facilities for OWL models, and therefore can serve as an attractive starting point for rapid application development. Developers can initially wrap their components into a Protégé tab widget and later extract them to distribute them as part of a stand-alone application.

The Protégé-OWL editor enables users to:

- Load and save OWL and RDF ontologies.
- Edit and visualize classes, properties, and SWRL (Semantic Web Rule Language) rules.
- Define logical class characteristics as OWL expressions.
- Execute reasoners such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup.

The Protégé API not only has a non-visual model part, but also comes with comprehensive support for user interface programming. There are several convenient classes and utility methods that help programmers develop interactive user interfaces quickly and with uniform look-and-feels that match the rest of the Protégé family of tools.

One of the foundations of UI programming with Protégé is

Volume : 4 | Issue : 1 | Jan 2014 | ISSN - 2249-555X

the event mechanism, which allows programmers to react cleanly on changes[6].

- Protégé is a free, open-source platform to construct domain models and knowledge based applications with ontologies.
- Ontologies range from taxonomies, classifications, database schemas to fully axiomatized theories.
- Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce and web services.

There are two main ways of modeling ontologies [7]:

- Frame-based
- OWL

Each of them have its own user interface.

- Protégé Frames Editor: enables users to build and populate ontologies that are frame-based, in accordance with OKBC (Open Knowledge Base Connectivity Protocol).
    - Classes
    - Slots for properties and relationships
    - Instances for class

- Protégé OWL Editor: enables users to build ontology for the Semantic Web, in particular to OWL
    - Classes
    - Properties
    - Instances
    - Reasoning

III. DIFFERENCE BETWEEN JENA AND PROTÉGÉ

Jena is a general RDF/RDFS framework. Thus Jena lacks specific primitives for OWL-based applications. It is the opposite with the Protégé- OWL API which is dedicated to OWL manipulation and provides most functions needed to exploit OWL ontology. This results in a faster and simpler programming. The Protégé-OWL API is the final choice for the programming needs. It is worth noting that these API being Java-based, this implies at least the core of the applications to be coded in Java.

IV. CONCLUSION

Both are APIs and can be used for similar tasks with the only main difference that Protégé-OWL is based on a much older framed-based API which predates OWL and RDF. Therefore Protégé had to do many design compromise which was found inconvenient. Jena on the other hand has been designed for RDF and OWL from the start and is optimized for handling triples, queries, etc. The Protégé-OWL used Jena for parsing and provides a Jena "view" (implementation of the Graph interface) so that some Jena services can be exploited for Protége. It can be easy to create an ontology file with Protégé and read that into a Jena Model and then process it as required. The Choice of which API to used should be determined by two factors as given below:

- Does the API offer a set of programming abstractions that can be used comfortably to implement the user requirements.
- Is the API actively supported and bug fixed in case the standards changes.

REFERENCE [1] "W3C Semantic Web Frequently Asked Questions". W3C. http://www.w3.org/2001/sw/SW-FAQ. Retrieved March 13, 2008. | [2] Berners-Lee, Tim; James Hendler and Ora Lassila (May 17, 2001). "The Semantic Web". Scientific American Magazine. http://www.sciam.com/article.cfm?id=the-semanticweb& print=true. Retrieved March 26, 2008. | [3] Herman, Ivan (March 12, 2008). "W3C Semantic Web Activity". W3C. http://www.w3.org/2001/sw/. Retrieved March 13, 2008. | [4]JENAhttp://jena.sourceforge.net/tutorial/RDF_API/index.html. | [5]http://jena.sourceforge.net/tutorial/RDF_API/ | [6] Chaoqing Lv, Takashi Kobayashi, Kiyoshi Agusa, Kun Wu, Qing Zhu | Matthew Horridge,Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe: A Practical Guide To Building OWL Ontologies Using The Protégé- OWL | [7]Tutorial:http://protege.stanford.edu/conference/2005/slides/T2_OWLTutorialI |