

Virtuoso View of Semantic Web



Anthoniraj Amalanathan, Vairamuthu Subbiah, Magesh Vasudevan

Abstract: Semantic web is an extension of the World Wide Web (WWW) which makes the machine to understand the content of the web. The main idea behind the semantic web is to share the interlinked documents over Internet in a meaningful way. Web contents are created and designed to be read by humans that may not be understood by computer programs. Thus, they fail to communicate with each other. Semantic web brings the web structure to meaningful form of information. This article describes building blocks, several tools and techniques to implement a killer application on the semantic web.

Keywords: Semantic Web, HTML, XML, RDF, Meta data, Microformat, Ontology.

I. INTRODUCTION

XML is the base of Semantic Web. The semantic content can be generated for search engines or domain specific applications[1]. In case of search engine's perspective, the content in HTML is associated with the concepts defined in the ontology, and the search engines should transform such content for indexing, in the same way as the meta-data is used to add extra information about pages while indexing [2].Such representation of HTML with RDF is termed as RDFa. Further, the triple stores can be used to push RDF triples defined over ontology and queried in accordance with application specific requirements[3]. Figure 1 shows the blocks over which the semantic web is built on.

A. Meta Data – The Beginning

Metadata is used to provide additional information about the web page [4]. It can be related to the meaning of the web page content. The sample HTML web page in Script 1 describes the person Anthoniraj who is working as an Associate Professor in Vellore Institute of Technology. The body contains the information of the person which will be displayed in the Web Browser. The header part of the page contains Metadata such as keywords, description of the page, and last modification date. This information can be used by a search engine or agent to retrieve meaningful content [5].

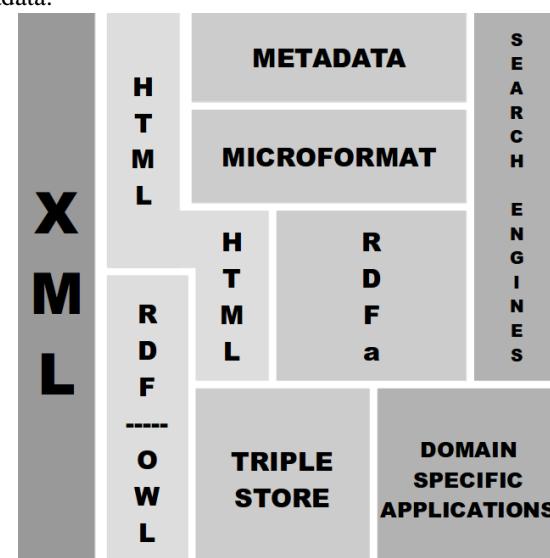


Fig. 1. Building Blocks of the Semantic Web.

```
<!DOCTYPE html>
<html>
<head>
    <title>Metadata Example</title>
    <meta charset="utf-8">
    <meta name="keywords" content="Metadata, Semantic Web" />
    <meta name="description" content="Understanding the Semantic Web - Metadata." />
    <meta name="revised" content="anthoniraj.com, 10/02/2019" />
</head>
<body>
    <section>
        <ul>
            <li>Anthoniraj Amalanathan</li>
            <li>Assistant Professor</li>
            <li>Vellore Institute of Technology</li>
            <li>aanthoniraj@vit.ac.in</li>
            <li>http://www.anthoniraj.in</li>
        </ul>
    </section>
</body>
</html>
```

Script 1. Sample HTML Script for a Person

C. Microformat

Microformat is a semantic markup to re-use existing HTML tags to convey metadata. Anyone parsing your website can use the microformat to find relevant information. There are several open microformat draft standards suitable for embedding data in HTML5.

- **h-card** is a simple, open format for publishing people and organizations on the web.



Manuscript published on November 30, 2019.

* Correspondence Author

Anthoniraj Amalanathan*, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

Vairamuthu Subbiah, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

Magesh Vasudevan School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Virtuoso View of Semantic Web

- (<http://microformats.org/wiki/h-card>)
- *h-recipe* is a simple, open format for publishing recipes on the web.
(<http://microformats.org/wiki/h-recipe>)
- *h-resume* is a simple, open format for publishing resumes and CVs the web.
(<http://microformats.org/wiki/h-resume>)
- *h-review* is a simple, open format for publishing reviews on the web.
(<http://microformats.org/wiki/h-review>)
- *h-geo* is a simple, open format for publishing geographic ordinates.
(<http://microformats.org/wiki/h-geo>)
- *h-event* is a simple, open format for publishing events on the web.
<http://microformats.org/wiki/h-event>

The following HTML web page contains *hcard* microformat which describes a person and his organization.

```
<!DOCTYPE html>
<html>
<head>
  <title>Microformat Example</title>
  <link rel="profile" href="http://microformats.org/profile/hcard">
</head>
<body>
  <section>
    <ul class="vcard">
      <li class="fn">Anthoniraj Amalanathan</li>
      <li class="note">Associate Professor</li>
      <li class="org">Vellore Institute of Technology</li>
      <li class="email">aanthoniraj@vit.ac.in</li>
      <li class="url">http://www.anthoniraj.com</li>
    </ul>
  </section>
</body>
</html>
```

Script 2. hcard information about a person

Microformat libraries are available in most of the programming languages. The Python parser script is given for demo to retrieve values from HTML page based on microformat attributes.

```
import mf2py
with open('2-microformat.html', 'r') as file:
    obj = mf2py.parse(doc=file)
    #print obj['items'][0]['properties']
    print obj['items'][0]['properties']['name'][0]
    print obj['items'][0]['properties']['email'][0]
    print obj['items'][0]['properties']['url'][0]
```

Script 2. Python Script to Parse Microformat

II. XML - THE BACKBONE OF THE SEMANTIC WEB

XML is the communicating language between machines; the structure of the information is given in the XML schema [6]. Though the XML schema can represent the structure, it fails to represent the meaning of the content. The XML schema is a World Wide Web Consortium (W3C) standard, Unicode data can be embedded in any XML document. A single XML document can refer to the elements from multiple schemas, where the namespace comes into a role of identifying the definition of the element. The namespace URI usually represents the location of XML schema.

```
<?xml version='1.0'?>
<description
  xmlns:p="http://www.example.com/person.xsd"
  xmlns:u="http://www.example.com/university.xsd">
```

```
<p:person>
  <p:name>
    <p:first>Anthoniraj</p:first>
    <p:last>Amalanathan</p:last>
  </p:name>
  <p:age>36</p:age>
  <p:work>
    <p:role>Assistant Professor</p:role>
  <p:organization>
    <u:university>
      <u:name>Vellore Institute of Technology</u:name>
      <u:loc>Vellore </u:loc>
      <u:web>vit.ac.in</u:web>
    </u:university>
    <p:exp>14</p:exp>
  </p:organization>
  </p:work>
</p:person>
</description>
```

Script 3. XML Document with Schema

It is necessary to follow four standard steps to create a schema for a particular XML document.

- Create / Use XML Document(info.xml)
- Set Constraints / Rules for your XML Document Rules
- Create XML Schema File (person.xsd)
- Link the schema file in your XML root element.

```
<?xml version="1.0"?>
<!--
  Schema for namespace person (p)
-->
<xs:schemaxmlns:xs="http://www.w3.org/XMLSchema" >
  <xs:complexType name="pname">
    <xs:sequence>
      <xs:element name="first" type="xs:string" />
      <xs:element name="last" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="pwork">
    <xs:sequence>
      <xs:element name="role" type="xs:string" />
      <xs:element name="organization" type="xs:anyType" />
    </xs:sequence>
  </xs:complexType>
<?xml version="1.0"?>
<!--
  Schema for namespace university (u)
-->
<xs:schemaxmlns:xs="http://www.w3.org/XMLSchema" >
  <xs:element name="person" minOccurs="1"
    maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="pname" />
        <xs:element name="work" type="pwork" />
        <xs:element name="email" type="xs:string" />
        <xs:element name="url" type="xs:string" />
        <xs:element name="exp" type="xs:integer" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="university" minOccurs="1"
    maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string" />
        <xs:element name="loc" type="xs:string" />
        <xs:element name="web" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:element>
```

</xs:element>
</xs:schema>

Script 4. XML Schema Definition

Element is a key part of XML schema. It is divided into two parts, such as simple and complex. Data type (Integer, String, Double, Boolean, and Date) should be specified for each and every element. A complex element may contain a set of child elements and attributes. Figure 2 describes the conditions to be considered while creating XML Schema.

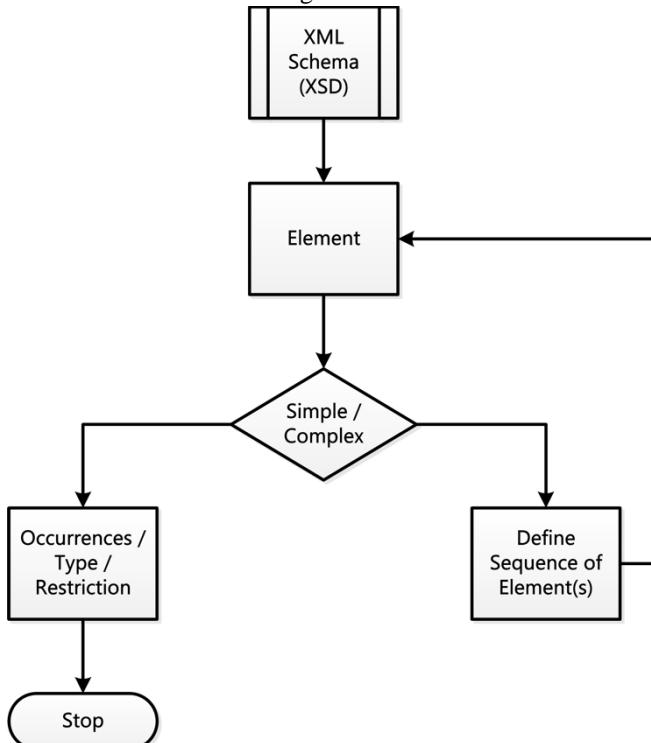


Fig. 2. Flow Diagram for creating XML Schema.

III. ONTOLOGY

The ontology plays an important role in delivering the semantics of the content. Ontology can be used to build systems in three ways as follows,

- A large knowledge base of concepts expressing the existence, which can be used by an Artificial Intelligent system to represent its knowledge about all. The system is designed to understand the concepts and make it applicable for interacting, learning and inferring knowledge in a perceived environment. DBpedia and YAGO are knowledge sources with auto-generated large Ontology from Wikipedia content.
- A domain-dependent Knowledge source, where the system is pertained to understand the concepts of that particular domain and interact with the users based upon it. The closed domain systems are most feasible where the boundary of understanding is limited with the concepts of the domain, and it can be efficiently managed. The advantage of knowledge sources using ontology is the flexibility of extending or changing the schema.
- Web is a large source of knowledge where resources are defined, produced and interlinked by every organizations.

The Ontology-based systems must be designed in a manner that the actions of the systems can be performed based upon the concepts. Thus, the system knows how to handle or react

to certain information. This facilitates system to system interaction that is agreed upon using a common language as ontology of open or closed domain.

OWL is a standard of W3C to express the ontology in XML form. These concepts are referred by the resource in the RDF document. Let us look into the case with a very simple ontology for defining a person, organization and area of interest which is shown in Figure 3. These concepts are referenced in RDF when defining the individuals, which is explained in the next section.

A. Web Ontology Language (OWL)

OWL is a structured representation to define an Ontology whose standards are defined by W3C [7]. Using OWL, we can define the various forms of concepts that can be expressed using description logics. Description logic terms like the concept are analogous to class, instance as individual and relationship as a role. OWL can define concepts with cardinality, characteristics and restrictions of the role. The reader is advised to have a further reading of defining Ontology through OWL syntax in W3C standard specification[8], [9].

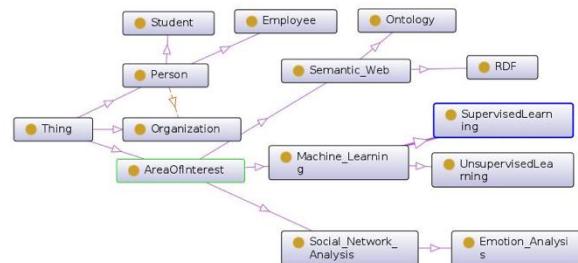


Fig. 3. Ontology graph for the case study.

B. Representing Knowledge Resources

There are some standards which can be used for describing knowledge in the semantic web. These standards are used according to the various requirements in web development. Following are the basic standard used in most of the semantic web knowledge representation [10]...?

IV. RESOURCE DESCRIPTION FRAMEWORK (RDF)

RDF advances the representation to description logic expressed in the form of ontology[11]. The resource in the RDF document refers to any concept in the ontology and utilizes the roles of ontology to a link between the concepts. The important role played by RDF is that linking of resources with uniqueness, and further conceptualizing the resources by linking to the ontology in an OWL or RDF/XML document. The previous representations capture only the structure of data, then validates it. In case of RDF, meaning of the data is captured, the resource identifies itself with a concept like book, person etc. The meaning of the concept is defined in the ontology, for example the concept book can be defined as group of chapters on a domain compiled by an author and published by a publisher, where author is a person and publisher is an

Virtuoso View of Semantic Web

organization[12]. Defining individuals for the ontology is given below.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ont="http://www.example.com/ontology">
  <rdf:Description about=
    "http://www.example.com/organization/VIT">
    <rdf:type resource=
      "http://www.example.com/ontology#Organization"/>
    <ont:name>
      Vellore Institute of Technology
    </ont:name>
    <ont:partOfOrg
      resource="http://anthoniraja.com/organization/VIT">
  </rdf:Description>
  <rdf:Description
    about="http://www.example.com/person/anthoniraj">
    <rdf:type
      resource="http://www.example.com/ontology#Faculty"/>
    <ont:name>
      Anthoniraja Amalanathan
    </ont:name>
    <ont:partOfOrg
      resource="http://anthoniraja.com/organization/VIT">
  </rdf:Description>
  <rdf:Description
    about="http://www.example.com/person/magesh">
    <rdf:type
      resource="http://www.example.com/ontology#Student"/>
    <ont:name>Magesh Vasudevan</ont:name>
    <ont:partOfOrg
      resource="http://anthoniraja.com/organization/VIT">
  </rdf:Description>
```

Script 4. RDF Definition of a Person

RDF supports the uniqueness of the resource, suppose a person is defined with an URI "http://anthoniraja.org/anthoniraj" it can be utilized around the web to define the roles played by person in various knowledge sources, for example, government can use it to link with its citizen knowledge base, hospitals can use it to link with its patients records, e-commerce services can use it to represent a customer; now the complete data is linked by an identifier of the resource. If various knowledge sources wish to use their identifiers, yet the links can be enabled by defining owl:sameAs relation between the identifiers.

The web is a large source of knowledge, yet the data is not linked, the content of information is plain text, the machines cannot understand it, the machines cannot meaningfully retrieve the content unless we explicitly define how to retrieve it. The data must be linked, for example if there is a text containing word "Gandhi" it must be specified which "Gandhi" it is speaking about, the unique resource represented for the person Gandhi must be linked, unless the content identifies itself with a unique resource over the web and the unique resource identifies itself with a concept in Ontology, the meaning cannot be understood by machines. Now, instead of defining machines based upon scenario and data, we can program it to understand the Ontology; which leads to machine work on the meaningful content [13], [14].

A. Notation-3 or N3 or RDF Turtle

N3 format is a non-XML representation of the linked data. It consists of Subject Predicate and Object triples. The Subject should be any resource identifier; the predicate can be an RDF relation or Ontology relation, object can be any XML data type for data property or resource identifier in case of object property [15].

```
@prefix ad: <http://www.example.com/ns/addressbook#>
@prefix id: <http://www.example.com/ns/data#>
id:10000 ad:firstName "Anthoniraj".
id:10000 ad:lastNmae "Amalanathan".
id:10000 ad:email "aanthoniraj@vit.ac.in".
id:10000 ad:mobile "1234567890".
id:10000 ad:designation "Associate Professor".
```

```
id:10001 ad:firstName "Vairamuthu".
id:10001 ad:lastNmae "Subbiah".
id:10001 ad:email "svairamuthu@vit.ac.in".
id:10001 ad:mobile "0987654321".
id:10001 ad:designation ""Associate Professor"".
```

Script 4. Notation-2 Representation

V. INFORMATION RETRIEVAL

Now we have defined ontology and added some individuals; further, we need to find information from the knowledge source. Simple Protocol and RDF Query Language (SPARQL) is a query language to retrieve information from the semantic knowledge sources[16], [17]. The query itself is in the triple form which iteratively expands or restricts based upon the next triple in the query [1]. For example, let us consider retrieving all students with an area of interest as semantic web,

```
PREFIX ont:<http://www.example.com/ontology>
select ?student {
  ?student rdfs:type ont:Student.
  ?student ont:aoi:ont:Semanitic_Web.
}
```

Script 4. SPARQL Query-1

The query returns with URI of students, if the details of the students have to be retrieved,

```
PREFIX ont:<http://www.example.com/ontology>
select ?student,?name,?regno {
  ?student rdfs:type ont:Student.
  ?student ont:aoi:ont:Semanitic_Web.
  ?student ont:name ?name.
  ?student ont:reg_no ?regno.
}
```

Script 5. SPARQL Query-2

Further queries like faculty who handle class which is not of their area of interest,

```
PREFIX ont:<http://www.example.com/ontology>
select ?student,?name,?regno {
  ?student rdfs:type ont:Student.
  ?student ont:aoi:ont:Semanitic_Web.
  ?student ont:name ?name.
  ?student ont:reg_no ?regno.
}
```

Script 6. SPARQL Query-3

In this scenario, if the faculty handles Machine Learning, where his area of interest is stated as Artificial Intelligence, even this information will be retrieved as the Machine Learning is a subclass of Artificial Intelligence. It is recommended to refer more details about SPARQL in W3C standard specification [18]. These queries are executed in Apache Jena RDF Engine and appropriate results are also obtained according to Ontology.

VI. RESULTS

To analyze the retrieval process, the Person and Student ontologies are imported into Apache Jena Fuseki and validated against various SPARQL queries (Script 1,2,3),



The screenshot shows the Apache Jena Fuseki interface. At the top, it says "Dataset: /semanticweb". Below that are buttons for "query", "upload files", "edit", and "info". A blue bar indicates "Available graphs" and "default graph (9 triples)". The main area shows a code snippet representing an RDF triple store:

```

1 @prefix ont: <http://www.example.com/ontology> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#
3 <http://www.example.com/organization/VIT>
4 a <http://www.example.com/ontology#Or
5 ont:name "<\n\t\t\tVellore Institute of Techn
6 ont:partOfOrg <http://anthoniraja.com/organizatio
7
8 <http://www.example.com/person/anthoniraj>
9 a <http://www.example.com/ontology#Pa
10 ont:name "<\n\t\t\tAnthoniraja Amalanathan\n
11 ont:partOfOrg <http://anthoniraja.com/organizatio
12
13 <http://www.example.com/person/magesh>
14 a <http://www.example.com/ontology#St
15 ont:name "Magesh Vasudevan" ;
16 ont:partOfOrg <http://anthoniraja.com/organizatio
17
18

```

Fig. 3. Search Results for Person Ontology

the search results clearly indicate that the retrieval is more effective than normal SQL query. The results can also be tuned in an effective way based on the domain parameters and schema.

VII. CONCLUSION

The semantic web is not built by a single organization or a community; it is transforming the knowledge in web to next level of machine interoperability where the human interpreted meaning is enabled in the machine interpretation of the same content. The standards and technology for the semantic web can enable developing closed domain applications which can work intelligently in case of multi-system environments where systems have to communicate the perceived environment and further make decisions based upon the meaning of the content. The need and ways to develop one such application is the aim of this paper, where a clear route map to building applications using the Ontology and RDF is explained.

REFERENCES

- [D. Beckett and B. McBride, “RDF/XML syntax specification (revised),” W3C Recomm., vol. 10, 2004.
- I. Hickson and others, “HTML5 specification,” W3C HTML Work Group, Febr., 2011.
- B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, “RDFa in XHTML: Syntax and processing,” Recomm. W3C, vol. 7, 2008.
- O. Lassila, “Web metadata: A matter of semantics,” IEEE Internet Comput., vol. 2, no. 4, p. 30, 1998.
- M. Nilsson and M. Palmér, “Semantic Web meta-data for e-learning: Some architectural guidelines,” in 11th World Wide Web Conference (WWW2002), 2002.
- D. C. Fallside and P. Walmsley, “XML schema part 0: primer second edition,” W3C Recomm., vol. 16, 2004.
- D. L. McGuinness, F. Van Harmelen, and others, “OWL web ontology language overview,” W3C Recomm., vol. 10, no. 10, pp. 2003–2004, 2004.
- S. Bechhofer, “OWL: Web ontology language,” in Encyclopedia of Database Systems, Springer, 2009, pp. 2008–2009.
- S. V. Jambhulkar and S. J. Karale, “Semantic Web application generation using Protégé tool,” in 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1–5.
- J. F. Sowa, “Knowledge representation: logical, philosophical, and computational foundations,” Pacific Grove, CA, Brooks/Cole, vol. 594, 2000.
- V. A. Chakkarwar and A. A. Joshi, “Semantic web mining using RDF

- data,” Int. J. Comput. Appl., vol. 975, p. 8887, 2016.
- D. Moussallem, M. Wauer, and A.-C. N. Ngomo, “Machine translation using semantic web technologies: A survey,” J. Web Semant., vol. 51, pp. 1–19, 2018.
- A. Amalanathan, S. Shanmugam, M. Senthilnathan, and S. Harikumar, “Semantic annotations for customizing geographical maps based on WordNet,” Procedia Technol., vol. 4, pp. 375–383, 2012.
- C. Bizer, “The emerging web of linked data,” Intell. Syst. IEEE, vol. 24, no. 5, pp. 87–92, 2009.
- S. Dietzold and S. Auer, “Access control on RDF triple stores from a semantic wiki perspective,” in ESWC Workshop on Scripting for the Semantic Web, 2006.
- B. Quilitz and U. Leser, Querying distributed RDF data sources with SPARQL. Springer, 2008.
- M. Arenas, G. Gottlob, and A. Pieris, “Expressive languages for querying the semantic web,” ACM Trans. Database Syst., vol. 43, no. 3, p. 13, 2018.
- S. Harris, A. Seaborne, and E. Prud’hommeaux, “SPARQL 1.1 query language,” W3C Recomm., vol. 21, 2013.

AUTHORS PROFILE



Anthoniraj Amalanathan is an Associate Professor at Vellore Institute of Technology (VIT) in School of Computer Science and Engineering. He is also a Deputy Director of Software Development Centre (SDC) at VIT, Vellore, India. Anthoniraj's areas of expertise include Semantic Web, Feature Engineering, Text Mining, Machine Learning, and Open Source Programming. His research interests

are in the use of technology in education and developing open source software that takes into consideration the unique needs of learners. He is currently doing research on Language Modelling and Chatbot to help students in Academic Activities. He received his Ph.D. in Computer Science and Engineering from VIT.



Vairamuthu Subbiah is associated with School of Computer Science and Engineering as Associate Professor at Vellore Institute of Technology (VIT). His major areas of research include Machine Learning, Internet of Things, Software Engineering and Open Source Programming. His research interests are in the use of technology in agriculture and under water communications to move forward in

smart farming and preserving various species that lives under water. He is currently doing research on identifying sleeping patterns of a mobile user to optimize various activities involved. He is having very good number of publications in various journals of national and international repute.



Magesh Vasudevan is a MTech graduate from VIT, Vellore. He is a passionate AI researcher, where knowledge representation and mining is his core work. He has worked on many projects where knowledge extraction is done through semantic means. He has also worked on natural language processing system where the knowledge is represented in semantic ontology and expressed in RDF. He is passionate about bringing machine learning into extraction of knowledge from the represented RDF and further process actions based on logics presented.