



TBL2RDF: HTML Table to RDF Translator



V.Kiran Kumar¹, K.Rajasekhara Rao²

¹Computer Science Department

Dravidian University

Kuppam – 517 425, India

kirankumar.v@rediffmail.com

²Koneru Lakshmaiah College of Engineering

Green Fields, Vaddeswaram 522502

Guntur Dt., Andhra Pradesh

India

krr_it@yahoo.co.in

ABSTRACT: *There is a large quantity of data available on the web, which is designed in HTML table format. Due to lack of machine-understandable information, these data cannot be processed by machines. This paper describes a TRANSLATOR, a tool designed as Java application to allow anyone, even non-experts, to design their data in HTML table format for use on the Semantic Web. This is accomplished by automatically translating data written in HTML table format into the Resource Description Framework Schema (RDFS). XML representation in RDF has several advantages, among them is compatible with existing Semantic Web standards.*

Keywords: World Wide Web (WWW), Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), Ontology.

Received: 14 October 2008. Revised 19 November 2008. Accepted 2 December 2008

© 2009 D-line. All rights reserved.

1. Introduction

The goal of the semantic web is that web-based agents can process and “understand” data rather than merely display them as present. There is a large quantity of data available in the web, which is designed with HTML tables. Due to lacking of machine understandable information, these data cannot be used by web-based agents or machines. RDF is a language representing resources on the web, which can be processed by machines rather than just displaying them. Hence, a tool was designed which extends syntactic HTML tables into RDF files. The main advantage of using this tool is that, a layman with little knowledge on RDF can easily create RDF documents by simply designing his data in HTML table format.

There is a number of automated conversion tools available to convert application specific format into RDF to use RDF tools and for integration with other data, such as D2RQ, java2rdf, javadoc2rdf, Excel2rdf etc., but the tool which is presented here is designed as a java application and it takes HTML file containing table as input and produces RDFS ontology. As there is much data available on web pages in a table format, this tool will be helpful for converting said data into RDF ontology.

The rest of the paper is organized as follows: Section 2 describes a brief description about Resource Description Framework (RDF) and RDFS. Section 3 discusses HTML tables. Section 4 discusses the tool architecture and its prototype. Section 5 explains the tool with a scenario and finally Section 6 concludes and explains the future work of the tool.

2. URI Ref and Namespaces

A Uniform Resource Identifier (URI) is a character string that identifies an abstract (or) physical resource on the web. A URI [5] reference (URI ref) is a URI with an optional fragment identifier attached to it and preceded by character “#”. For example, the following string is a URI ref [5], where “http://www.dravidianuniversity.ac.in” is a URI and “All_Genders” is a fragment identifier which was preceded by “#”.

“http://www.dravidianuniversity.ac.in#All_Genders”

A name space [5] is a collection of names. A name space is identified by URI ref. Names from namespaces are called as qualified names (Qnames) for example, ‘p:u’, where ‘p’ is the namespace prefix and ‘u’ is the local part. For example, ‘rdf:description’ is a qualified name where ‘rdf’ is namespace prefix and ‘description’ is a local part. This tool uses a set of namespaces, URI refs and Qnames for converting syntactic HTML table into semantic files.

3. RDF and RDFS

The World Wide Web (WWW) was originally built for human consumption, and everything on it is *machine-readable*, these data is not *machine-understandable*. The solution is to use metadata (data about data) which can be processed by machines. Resource Description Framework (RDF) [8] is particularly intended for representing metadata about web resources, which can be processed by applications rather than being displayed to people. In February 2004, The World Wide Web Consortium released the RDF as W3C recommendation. RDF is a general method to decompose knowledge into small pieces, having certain rules of semantics. The triples can be represented as subject, predicate and object. In RDF, the English statement

“Tim Berners-Lee invented World Wide Web”

Could be represented by RDF statement having

- A subject Tim Berners-Lee.
- A predicate invented.
- And an object World Wide Web.

RDF-Schema [5][9] (RDF-S) was introduced and recommended by W3C in Feb 2004 due to the lack of defining application-specific classes and properties in RDF. RDF Schema provides the framework to describe application-specific classes and properties.

3.1 Class: A class in RDF Schema is somewhat similar to ‘class’ in object oriented programming languages. In RDF Schema, a class is any resource having an ‘rdf:type’ property with the value of the resource ‘rdfs:Class’. And RDF classes can be used to represent almost everything on the web.

3.2 Property: A property is any instance of the class ‘rdfs:Property’. The ‘rdfs:domain’ property is used to indicate that a particular property applies to a designated class and ‘rdfs:range’ property is used to indicate that the values of a particular property are instances of a designated class (or) are instances of an XML schema data type.

3.3 Instance: An instance of a class C is a resource having an ‘rdf:type’ property whose value is C. A resource may be an instance of more than class.

4. HTML Tables

Table represents relationships between data. Before the creation of HTML table model, the only method available for relative alignment of text is through the use of PRE element. While this was useful in some situations, the effects of PRE are very limited. Tables were introduced in HTML 3.0 [1]. Since then a great deal of refinement has occurred. Fig.1 represents the structure of a table. The table may have an associated caption that provided short description of table’s purpose. The <table> tag is used to design data in a table format in HTML. A table is divided into rows (with <tr> tag) and each row is divided into data cells (with <td>, <th> tags). A Data cell may contain textual data as well as non-textual data (i.e., images, lists etc). In specific situations a column or row should have a “title”, for that <th> tag can be used to create headers for either rows or columns, or both. In fig.1 “table header” represents the column header and “stub” represents the row header which is defined by <th> tags. The table may contain optional “footnotes”.

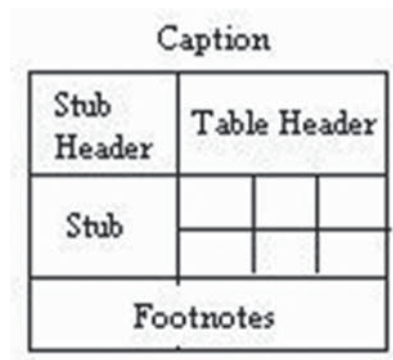


Figure 1. A structure of a table

There is a number of other tags and attributes available for designing the data in a tabular format, but the tool recognizes only said tags and remaining are ignored. At present, the tool recognizes the input file when all the following conditions are satisfied.

1. Table cells should not span more than one column or row.
2. Table cells must contain textual data only.
3. The table must have a table header.
4. The table may have a caption, stub, stub header and optional footnotes.

5. The Translation Process

A class in RDF Schema is somewhat like a notion of a ‘class’ in object-oriented programming languages. This allows resources to be defined as instances of classes and subclasses of classes. The TBL2RDF tool is a straightforward process and is illustrated in fig.2. The caption of a table is translated as a class name in RDFS file. If the table does not have a caption, then tool takes ‘TABNAME’ as a class name in creating an RDFS file. Similarly, the tool extracts rows of a table and in turn, takes its cell values to generate triples, using row-name or stub as ‘subject’, column name as ‘predicate’ and its cell value as ‘object’. The row-name or stub may not exist for some tables. In that case, the tool creates a blank node with node ID, which is treated as subject. This plays a crucial role in querying the file. This process is illustrated in fig.3.

Each table row produces a cluster of RDF triples excluding the first row. As the first row of a table must have the table header, which in turn, is used as a predicate in RDF triple. For each predicate, the domain name will be the class-name and label will be the cell value. For example, as shown in Fig. 2, the first row contains { Name, Height, Weight, Color }. These

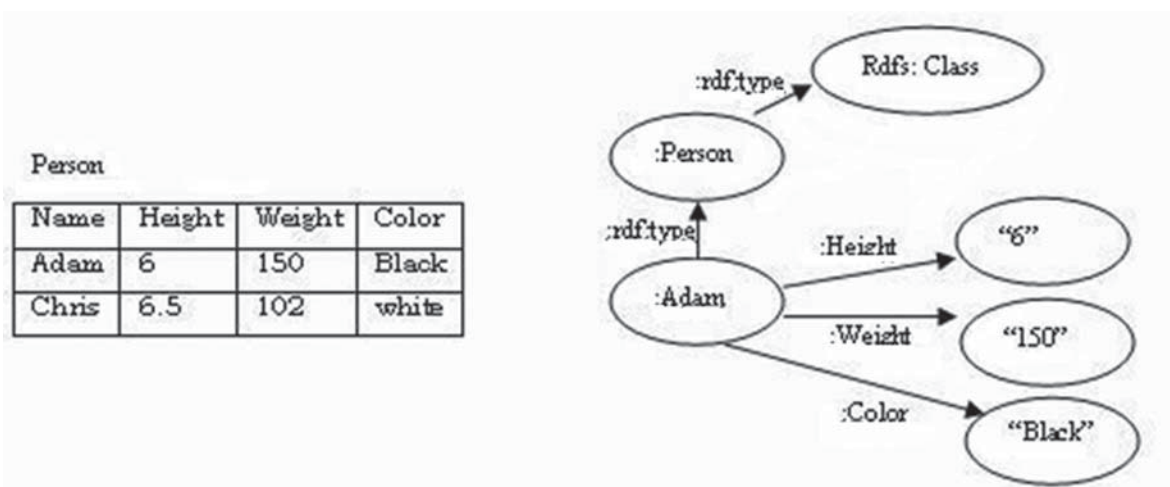


Figure 2. Translation of HTML table into RDF

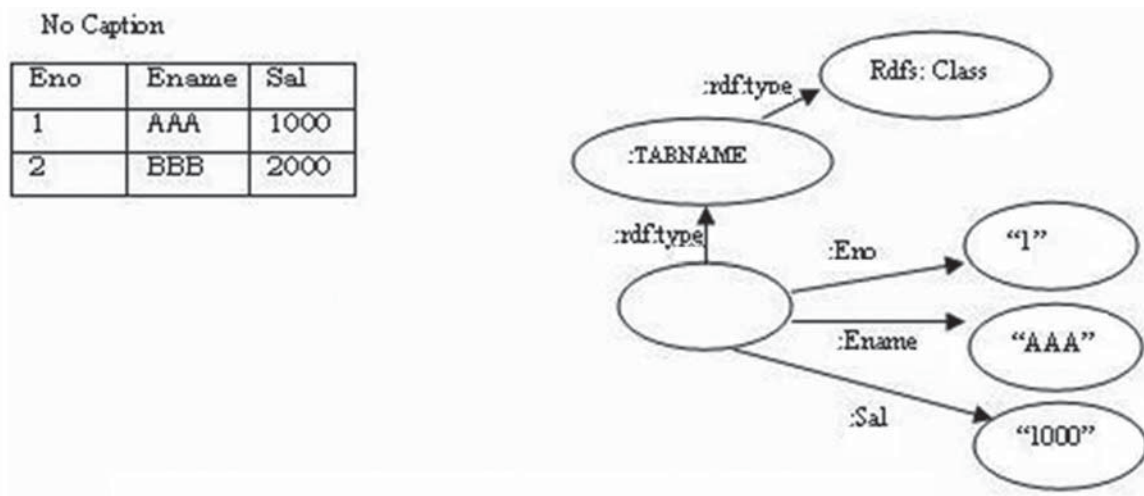


Figure 3. Translation of HTML table into RDF Blank Node

values are translated as predicates, for each predicate the domain name will be class-name. The other rows of a table are translated as instances of a class. Each row produces a cluster of RDF triples <subject, predicate, object> like { “Adam”, “Height”, “6” }, { “Adam”, “Weight”, “150” } etc., As a stub is not used in this table, the RDFS creates a blank node with ID as shown in Fig. 3.

5.1 TBL2RDF Tool

Fig.4 represents an overview of the TBL2RDF tool process. The translation process is taken into two modules namely ‘parser’ and ‘converter’. The ‘parser’ module, first scans the input file and returns the caption and size of a table i.e., no of rows and columns. If the table does not have a caption, then the ‘parser’ module assigns ‘TABNAME’ as a caption of a table. Based on the number of rows and columns of a table, the ‘parser’ creates a string array and assigns each cell with cell value of a table, no matter whether the data cell is represented as either <td> or <th>.

The Procedure for the ‘parser’ module is given below. The procedure takes the input file and produces a two dimensional string array.

Procedure parser(caption, rowsize, colsize)

(variables used in procedure :

data[rowsize][colsize] as string array, row, col as integers)

1. Open input file
2. read tag from file
3. if the tag is <table> then repeat step until </table> tag
 - 3.1. read a tag from file
 - 3.2. if the tag is <tr> repeat step until </tr>
 - 3.2.1. row=row+1
 - 3.2.2. read a tag from file

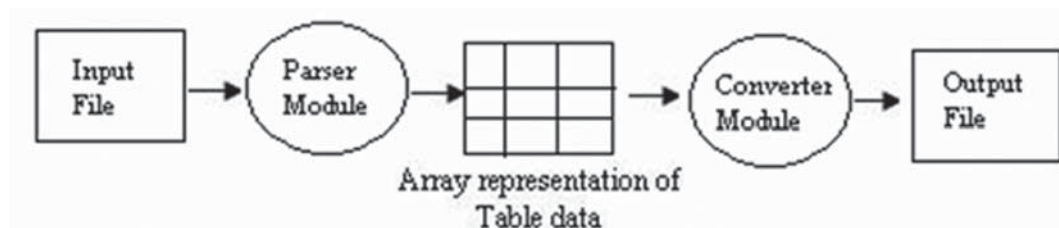


Figure 4. An overview of the Tool Process

- 3.2.3. if it is <th> or <td> then
 - 3.2.3.1. col=col+1
 - 3.2.3.2. data[row][col] = tag data
 - 3.2.3.3. goto 3.2.2
4. close input file

The 'converter' module uses array, produced by 'parser' module in creating RDFS ontology. This module requests the user to enter 'URI' and 'Prefix' variables for namespace declaration in RDFS file. Then, it asks the user whether the table has a stub or not. If yes, then first column of an array is considered as 'subject' of RDF triple. Otherwise, a blank node with node ID is created and considered as subject for RDF triple. The procedure for 'converter' module is given below. This procedure takes array as input and produces RDFS file.

Procedure converter(data[[]], caption)
 (variables used in procedure : URI, Prefix, ch)

1. open output file
2. read URI and Prefix
3. write 'The table has a stub or not [y/n] :'
4. read ch
5. write namespace specifications to output file
6. create an RDFS class with tablename, then write into output file
7. Read first row of an array and each cell is translated as property of RDFS, for each property the domain name will be class-name('caption')
8. i <- 1
9. repeat step until i<rowsize
 - 9.1 if (ch=='y') then subject = data[i][0]
 otherwise subject = s+""+i; // Blank Node with ID
 - 9.2 j <- 1
 - 9.3 repeat step until j<colsize
 - 9.4 predicate <- data[0][i], object <- data[i][j]
 - 9.5 create RDF triple with <subject, predicate, object>, then write into output file
 - 9.6 j <- j+1
 - 9.7 i <= i+1
10. write closing tags to output file
11. close output file

6. User Interface

Goals for designing the tool:

1. It should be accessible to all platforms.
2. User friendly.

Both goals will be satisfied as it is designed as a java application with a simple command line execution. The command line execution of a tool is shown in fig.5. The tool accepts input and output files as command-line arguments, then it requests the user to enter 'URI' and 'prefix' values for namespace creation before creating RDFS ontology. The third input is related to the table, if the table has a row-header or stub, then the row-header is treated as subject. Otherwise, a blank node with ID is created and treated as subject. Hence, the user must be careful while giving this input.

It exports data as RDF/XML serialization format in order to leverage other tools and code bases being built around XML. The array used in this particular tool has certain limitations. That is, it depends on the heap size of the java virtual machine (JVM). The default heap size of JVM varies depending upon an operating system. If the user needs large number, java provides a command, which helps in extending the heap size.



```
C:\WINDOWS\system32\cmd.exe

C:\RDFT00~1>java TBL2RDF Person.html Person.rdf
Enter Prefix :du
Enter URI :http://www.dravidian.in
Whether Table has stub or not [y/n] :y
Converted Successfully...

C:\RDFT00~1>_
```

Figure 5. Command-line Execution of Tool

Example Scenario – Person Table

Input: HTML Table

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<!-- Created with the CoffeeCup HTML Editor 2008 -->
<!--      http://www.coffeecup.com/      -->
<!--      Brewed on 2/19/2009 3:54:53 PM      -->
<head>
  <title></title>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
</head>
<body>
  <font size="2" color="#000000" face="Arial">
  <div><table width="100%" border="1" cellpadding="2" cellspacing="2" bgcolor="#ffffff">
  <tr valign="top">
  <td>Name<br></td>
  <td><br>Height</td>
  <td><br>Weight</td>
  <td>Color</td></tr>
  <tr valign="top">
  <td>Adam<br>
  </td><td>6<br></td>
  <td>150<br></td>
  <td>blue<br></td>
  </tr>
  <tr valign="top">
  <td>Chris<br></td>
  <td>5.5<br></td>
  <td>100<br></td>
  <td>white<br></td>
  </tr>
  </table>
  </div>
  </font>
</body>
</html>
```

Output: RDFS File

```
<?xml version="1.0"?>
<rdf:RDF xmlns:du="http://www.dravidian.in#" xmlns:owl="http://www.w3.org/2002/07/owl#"
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:about="http://www.dravidian.in#TABNAME">
</rdfs: Class>
    <rdf:Property rdf:about="http://www.dravidian.in#Name">
      <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
      <rdfs:label>Name</rdfs:label>
    </rdf:Property>
    <rdf:Property rdf:about="http://www.dravidian.in#Height">
      <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
      <rdfs:label>Height</rdfs:label>
    </rdf:Property>
    <rdf:Property rdf:about="http://www.dravidian.in#Weight">
      <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
      <rdfs:label>Weight</rdfs:label>
    </rdf:Property>
    <rdf:Property rdf:about="http://www.dravidian.in#Color">
      <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
      <rdfs:label>Color</rdfs:label>
    </rdf:Property>
    <rdf:Description rdf:about="du:Adam">
      <du:Height rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">6</du:Height>
      <du:Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">150</du:Weight>
      <du:Color rdf:datatype="http://www.w3.org/2001/XMLSchema#string">blue</du:Color>
    </rdf:Description>
    <rdf:Description rdf:about="du:Chris">
      <du:Height rdf:datatype="http://www.w3.org/2001/XMLSchema#float">5.5</du:Height>
      <du:Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">100</du:Weight>
      <du:Color rdf:datatype="http://www.w3.org/2001/XMLSchema#string">white</du:Color>
    </rdf:Description>
  </rdf:RDF>
```

Test Results

The World Wide Web Consortium (W3C) service was created by Nokia's Art Barstow (a former W3C Team member) and it is a free service available on the web to validate RDFS documents. The output produced by the tool was validated with W3C validator and the result is shown in the Fig. 6. Triples of the data model in N-Triples format, which was generated by the W3C Validator is shown in Fig. 7.

7. Conclusion and Future Work of the Tool

The primary advantage of this tool is that the existing web data which is in HTML table format can be easily converted into RDFS ontology. Secondary, a layman with little knowledge on RDF can easily create RDFS documents by simply designing his data in HTML table format. There are many limitations to the tool.

1. It recognizes textual data only (excluding lists). Research is being carried on to see how best non-textual data can also be taken for such conversion.
2. It does not recognize nested tables and more than one independent table in HTML file.
3. The tool recognizes integer, character, float and string data types of table data only. For other data types, the tool takes it as a string data type.

http://www.w3.org/RDF/Validator/ARPServlet

W3C[®] **RDF** Powered by **Validation Service**

Home Documentation Feedback

Validation Results

Your RDF document validated successfully.

The original RDF/XML document

```

1: <?xml version="1.0"?>
2: <rdf:RDF xmlns:du="http://www.dravidian.in#" xmlns:owl="http://www.w3.org/2002/07/owl#"
3:   <rdfs:Class rdf:about="http://www.dravidian.in#TABNAME">
4:     </rdfs:Class>
5:     <rdf:Property rdf:about="http://www.dravidian.in#Name">
6:       <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
7:       <rdfs:label>Name</rdfs:label>
8:     </rdf:Property>
9:     <rdf:Property rdf:about="http://www.dravidian.in#Height">
10:      <rdfs:domain rdf:resource="http://www.dravidian.in#TABNAME"/>
11:      <rdfs:label>Height</rdfs:label>
12:    </rdf:Property>

```

Jump To:

- Source
- Triples
- Messages
- Graph
- Feedback
- Back to Validator
- Input

Figure 6. W3C validator output

W3C[®] **RDF** Powered by **Validation Service**

Home Documentation Feedback

Validation Results

Your RDF document validated successfully.

Triples of the Data Model in N-Triples Format (Sub, Pred, Obj)

```

<http://www.dravidian.in#TABNAME> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.dravidian.in#Name> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.dravidian.in#Name> <http://www.w3.org/2000/01/rdf-schema#domain> <http://www.dravidian.in#TABNAME>
<http://www.dravidian.in#Name> <http://www.w3.org/2000/01/rdf-schema#label> "Name" .
<http://www.dravidian.in#Height> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.dravidian.in#Height> <http://www.w3.org/2000/01/rdf-schema#domain> <http://www.dravidian.in#TABNAME>
<http://www.dravidian.in#Height> <http://www.w3.org/2000/01/rdf-schema#label> "Height" .
<http://www.dravidian.in#Weight> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.dravidian.in#Weight> <http://www.w3.org/2000/01/rdf-schema#domain> <http://www.dravidian.in#TABNAME>
<http://www.dravidian.in#Weight> <http://www.w3.org/2000/01/rdf-schema#label> "Weight" .
<http://www.dravidian.in#Color> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.dravidian.in#Color> <http://www.w3.org/2000/01/rdf-schema#domain> <http://www.dravidian.in#TABNAME>
<http://www.dravidian.in#Color> <http://www.w3.org/2000/01/rdf-schema#label> "Color" .
<du:Adam> <http://www.dravidian.in#Height> "6" ^^http://www.w3.org/2001/XMLSchema#integer .
<du:Adam> <http://www.dravidian.in#Weight> "150" ^^http://www.w3.org/2001/XMLSchema#integer .
<du:Adam> <http://www.dravidian.in#Color> "blue" ^^http://www.w3.org/2001/XMLSchema#string .
<du:Chris> <http://www.dravidian.in#Height> "5.5" ^^http://www.w3.org/2001/XMLSchema#float .
<du:Chris> <http://www.dravidian.in#Weight> "100" ^^http://www.w3.org/2001/XMLSchema#integer .
<du:Chris> <http://www.dravidian.in#Color> "white" ^^http://www.w3.org/2001/XMLSchema#string .

```

Jump To:

- Source
- Triples
- Messages
- Graph
- Feedback
- Back to Validator
- Input

Figure 7. W3C validator output in notation3 (N3) Format

4. As the table data is moved into two dimensional string arrays, the limitation of the array depends on corresponding Java Virtual Machine (JVM).

Research is in progress to overcome the limitations regarding the tables.

References

- [1] Berners-Lee T, Hendler J, Lassila O. The Semantic Web, Scientific American, 2001.
- [2] Brickley D, Guha RV. Resource Description Framework (RDF) Schema specification 1.0: RDF Schema. W3C Working Draft, 2003.
- [3] Hayes P. Resource Description Framework (RDF) semantics. W3C Working Draft, 2003
- [4] Berners-Lee T. Weaving the Web – The Past, Present and Future of the World Wide Web by its Inventor. Texere, 2000.
- [5] Breitman K, Marco Antonio Casanova, Walter Truszkowski. Semantic Web, Concepts, Technologies and Applications 2006. 57–77.
- [6] Stephen Ferg. Bureau of Labor Statistics. Techniques for Accessible HTML Tables, August 23, 2002.
- [7] W3C Validation Service, <http://www.w3.org/RDF/Validator>, Feb 2007.
- [8] Manola F, Miller E. RDF primer, W3C Recommendation, 10 Feb, 2004. <http://www.w3c.org/TR/rdfprimer>.
- [9] RDF Vocabulary Description Language 1.0 : RDF Schema, W3C Recommendation 10 Feb 2004. <http://www.w3c.org/TR/rdf-primer>.
- [10] David Hausheer. Increase heap size in java to prevent java.lang. OutOfMemoryError, Nov 2007. <http://hausheer.osola.com/docs/5>.

Author Biographies



Kiran Kumar V received the BSc and MSc degrees in Computer Science in 1996 and 1998, respectively from Andhra University, Visakhapatnam and Acharya Nagarjuna University, Guntur. He is working as as an Assistant Professor of Computer Science at the Dravidian University, Kuppam. He is pursuing his PhD at Acharya Nagarjuna University. His research interests are Web Technology and Semantic Web. Currently, he focuses on translation of present web data which is in syntactic format into semantic format which can be executed by the machine.



Rajasekhara Rao Kurra received the MS and PhD degrees in 1992 and 2008, respectively, from B.I.T.S, Pilani and Acharya Nagarjuna University, Guntur. He is working as a Vice-Principal and Dean of KL College of Engineering, Guntur. His research interests are in Embedded Systems. He was attending many conferences and published several papers in international journals. He is acting as a software advisor for a software company at NBITS, Vijayawada. He was received “Best Teacher Award” 5 times from KL College of Engineering. Presently he is guiding four research scholars on different research areas.