Data Mapping for Transformation from RDB Schema to RDF Schema

K. R. Malik[1], T. Ahmad[2], M. M. Iqbal[3]

[1,2,3] Computer Science and Engineering Department, UET Lahore, Pakistan
[1]krmalik@gmail.com

Abstract-In this paper, we discussed the data mapping for transformation from relational database (RDB) schema to resource description frame (RDF) Schema. During transformation process between these two schemas, weaknesses like compatibility issues, update query and complexity in relationships are generated. We proposed an approach to overcome these issues particularly when data is transformed from RDB to RDF for semantic web applications. As, for evolving data keeping changes intact is hard and difficult to sustain. Main focus of this study is to map up common features found in both data models of RDB and Semantic Web (SW) based schemas using either form of XML as an intermediate which will help in improving transformation results. These data mappings can further help in gaining better compatibility options for data transformation.

Keywords-Data Transformation, Extensible Markup Language Schema (XMLS), Document Type Definition (DTD), Resource Description Framework Schema (RDFS), Relational Database Schema (RDBS), Data Type Mapping

I. INTRODUCTION

In Web applications, generally data is stored in the form of RDB. The web is semi-structured and unorganized in a formal way [i]. When it comes to searching and getting in touch with resources like web pages, peoples, and other web contents like video, audio etc. search engines work as a tool for web. Despite of advances made to these search engines, size of web content beats technology advancements. To overcome this problem, the Web content representation is required to be translated into machine-processable form. To translate such information into machine understandable form Semantic Web (SW) has been introduced [ii]. The need of SW increases due to its capability in providing improved methods and intelligent data seeking mechanisms and became a big evolution in the next generation of web [iii]. And to make this possible, old technologies are getting transformed into new ones. The EXtensible Markup Language (XML) based documents are getting enriched to map-up with Semantic Web [iv-vii]. Resource description framework (RDF) for the Semantic Web is a language to represent information on the internet in the form of triplets; subject, predicate and object [viii]. Which are used among resources mapped at hierarchal levels using graph based representations. Data types supported by XML play the key role in transformation to work properly. Whereas, customized data types can also be prepared using XML based tags [ix]. This is another reason due to which transformation made by different techniques and algorithm fails to support each other at their full level to support compatibility of data among systems. So, there is a need to look into different capabilities of data types supported by XML either by Document Type Definition (DTD) or XMLS along with their limitations.

II. STATE OF THE ART

The bulk of data and information are found on the Web is stored and retrieved using RDBs. Previous research shows that Semantic Web collaboration with other domains extends its utilization beyond the Web [iii]. Many methods and tools were introduced to help by providing the ways to explore relational data for availability to Semantic Web based systems [x]. Yet, there exist problems in clearly gaining results with high performance and compatibility [xi-xiii].

The standard which is known as XML document provides tags for the web. Where semi-structured and user defined, and predefine tags are stored. Though, the XML document is written either using XML schema or DTD [ix]. So, for transforming a relational schema into RDF schema, we have to do it partially as relational schema into XML schema and then XML schema into RDF schema or relational into DTD and then DTD into RDF schema. After that, transforming it back into its original form of relational schema.

Mapping can be done from RDB to RDF either by direct or indirect methods. Indirect methods like RDB to RDF Mapping Language (R2RML) involves table mapping from RDB to RDF without intermediate utilization of XML. Other indirect methods are about transforming schema from RDB to RDF using XML. XML schema can be either in the DTD or XML [xi-xiii]. Which is focusing on indirect methods based mapping.

Different researchers have attempted different approaches to fulfill the required outcome of reaching to a next generation web that is the Semantic Web. There have been attempts made for transforming a database schema into the Document Type Definition (DTD) and then into Resource Description Framework Schema (RDFS) either partially or completely. DTDs at the end of XML document for the web can be transformed and mapped into RDF schema. By looking into tags available in the XML document to pick better suitable tags for the satisfying role as a class or property [v]. On the other hand, XML documents can be updated to gain capability of being interpreted as RDF. It would be better if the XML's original structure remains unchanged and transformation process better results and coverage of the developed technique [vi, xiv-xvi].

In depth, the idea of resources linking among database, DTD and RDF is by using queries without changing context and meaning of web based data to form its semantic existence [xvii-xix]. These approaches can be captured to get a single side transformation approach. Other than these differences the major one is that each resource in RDF is assigned a unique Uniform resource identifier (URI) for identification. So, information on both of these semantic based expressions is given a shape of triples [xx].

This research can help to reduce complexity and compatibility issues for the process of transformation. As complexity and compatibility issues arise due to database schemas and ontologies are evolving at a constant speed to map with application and user requirements. Therefore, instead of mapping being redefined from scratch it should evolve on top [xiii]. The update statement concerning RDF stores is still under progress and its semantics are not yet well defined, and uncertainty remains concerned about the transformation of few SPARQL (SPARQL Protocol and RDF Query Language) Update statements. Only elementary (attribute-to-property & relation-to-class) mappings have been studied up to now. The problem of modifying relational data using SPARQL Update is the same as to view, update problem the classic database [xiii, xxi, xxii].

It is thought that one cause for the delay in the recognition of the Semantic Web is the deficiency of application and tools showing benefits of semantic web technology [xxv]. The success depends of large amount of data concerning semantic web of these tools [xxiv]. Because relational databases are considered as highly used medium for storage of data on web. The solution was to automatic manage mass of data in SW form as RDF.

It is studied that the transformation model from relational DB to RDF and storage mechanism of RDF stores in RDB. It is important to map existing relational representation among relational schema, DTD, XML schema, and RDFS [ix, xx, xxiii]. Proposed study of the above mentioned problem situation is possible through mapping of differences at schema level, enriching the DB contents on web based mapping results and identifying grey areas.

III. PROPOSED WORK

Transformation from one data model into another comes with its limitations. In this field of study, first look into mapping between different data models. Then the process of transformation using DTD and XMLS is given. A control experiment presented to show that how transformation works and on the basis of that results are given.

A. Mapping

A tabular representation in Table I to show the corresponding schema entity for the each concept of Database. These entities are further used for proper transformation among RDB and semantic web. In Table 1 each concept of a database is mapped with it one of the possible mapped solution in DTD, XML schema, RDFS. Table I shows how each entity of a database can be stored in its corresponding mapped field and command of other technologies like DTD, XML and RDF. Furthermore, some of the cases are discussed with the help of W3C standards for all of these technologies after careful study. The Table I can play a crucial role in transforming from database to semantic web and back to the database.

TABLE I
SHOWING ALTERNATIVE SCHEMA ELEMENTS FOR EACH CASE OF RELATIONAL SCHEMA VALUE

Concepts	Relational Schema	DTD	XML Schema	RDFS
Table	Table_Name	!ELEMENT Table_Name* ATTLIST !ELEMENT	Complex type element	Class
Field	Field_Name	!ELEMENT Field_Name	Element	Rdf: Property
Cardinality	Field(>=0)	!ELEMENT Field_Name*	Restriction (Pattern)	
Cardinality	Field(>0)	!ELEMENT Field_Name+	Restriction (Pattern)	
Referencing	Field	!ELEMENT Ref Key Field_Name* IDREF #REQUIRED	Simple type element Ref	Domain (Property)
Primary Key	Field	!ELEMENT Key Field_Name? ID #REQUIRED	Attribute Use="required"	Range
Composite key		!ELEMENT Second Key Field_Name? ID #REQUIRED	Attribute Use="required"	Rdfs:sub Property Of
Data type	Field	PCDATA or CDATA	Type	Type

In Table I concept of database covered are tabulated, field, cardinality, referencing, data types and keys. The elements are the basic entity used for both XML and DTD which are used for table and fields. Each element can further contain a list of attributes capable to map each field of a table from database. Keys are mapping is crucial as they play the main role in database for identification of records. So, each key is mapped corresponding term in XML and RDF. For transformation to happen according to defined mapping in this study have to follow mapping table. Which will give results of transformation from RDBS to RDFS and vice versa.

TABLE II
SHOWING CAPABILITIES OF TECHNOLOGIES USED IN TRANSFORMATION

Terminologies	RDBS	DTD	XML Schema	RDFS
Referential Identifier	✓	✓	✓	✓
Unique Identifier	✓	✓	✓	✓
Composite Unique Identifiers	✓		✓	✓
Enclosed Lists	✓	✓	✓	
Formal Semantics				✓
Inheritance				✓
Datatypes	✓		✓	✓
Constraints	✓		✓	
Cardinality constrains	✓		✓	

Whereas Table II shows different terminologies of the database can be mapped into a DTD, XML schema and RDFS. This table also shows each technology capability to support in transforming along with their limitations. For example, in case of DTD only available data type is equivalent to a string.

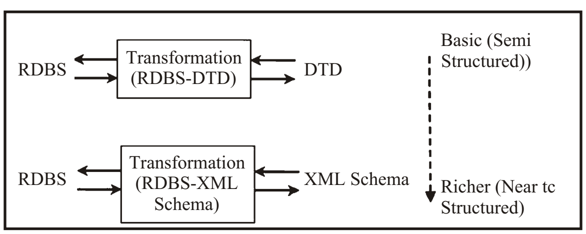

Fig. 1. Two approaches with structural level difference in Transformation Process

While trying to be more in control of data for the transformation (see Fig. 1) from RDBS to XML Schema and then back into RDBS. Then rich interface will be fine as long as all of this data passes through it to get transform up to the needed shape acceptable by the semantic web. Among semi-structured data which have a richer approach is better due to the high range of data type availability.

B. Experiment

To show the working of the elaborated process for transformation, first let's look at the RDBS of considered example of relational schema of the entity named "Organization" shown in Fig. 2. There are four tables Dependent, Employee, Manager and Department. We have used different tools like MS SQL Server 2005 for relational databases, Altova XMLSpy for XML Schema creation and an online project (X2R Converter) for RDF creation. As these tools are not meant to transform RDBS into RDF, so remaining experiment is done by coding in Java Programming Laguage to get results according to this research requirement.

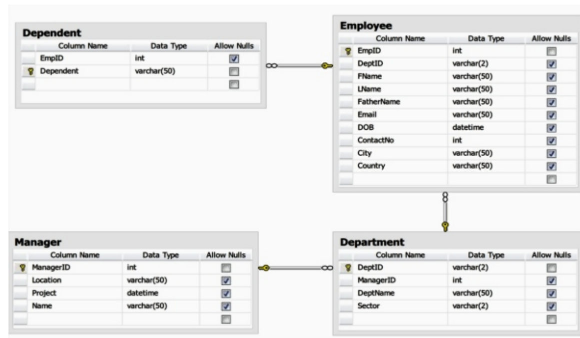

Fig. 2. Schema of RDB taken as example to show results of transformation process

Now by using Table I & III given in the Section III mappings rules for RDBS (shown in Fig. 2) is further transformed into Fig. A1 and Fig. A2, as a result, an intermediate format of web based technologies either like DTD or XML Schema is gained. Experiment is performed using tool of Altova XMLSpy.

In DTD transformed form of RDB (shown in Appendix Fig. A1) data type PCDATA is on covering the string data type of the database. In this much of the information is lost like corresponding data are of which type. Another problem in DTD which can be observed the lack of covering constraint on each data model. On the other hand XML Schema (shown in Appendix Fig. A2) is covering different data types and constraints.

TABLE III
XML PRIMITIVE DATA TYPES SUPPORTED FACETS

Datatypes \ facets	pattern	whitespace	Length	min Length	max Length	enumeration	total Digits	fraction Digits	min Inclusive	max Inclusive	min Exclusive	max Exclusive
string	✓	✓	✓	✓	✓	✓						
boolean	✓	✓										
decimal	✓	✓				✓	✓	✓	✓	✓	✓	✓
Float	✓	✓				✓			✓	✓	✓	✓
double	✓	✓				✓			✓	✓	✓	✓
duration	✓	✓				✓			✓	✓	✓	✓

dateTime	✓	✓				✓			✓	✓	✓	✓
Time	✓	✓				✓			✓	✓	✓	✓
Date	✓	✓				✓			✓	✓	✓	✓
gYear	✓	✓				✓			✓	✓	✓	✓
gMonthDay	✓	✓				✓			✓	✓	✓	✓
gDay	✓	✓				✓			✓	✓	✓	✓
gMonth	✓	✓				✓			✓	✓	✓	✓
haxBinary	✓	✓	✓	✓	✓	✓						
base64 Binary	✓	✓	✓	✓	✓	✓						
anyURI	✓	✓	✓	✓	✓	✓						
QName	✓	✓	✓	✓	✓	✓						

And by using XML Schema of RDB is transformed into XML file containing data of all tables according to their dependencies (shown in Appendix Fig. A3). Here according to the relationship based dependencies of one to many each instance of Manager Table in XML contains all dependent Department instances. Similarly, a Department instance contains all Employees of that department. And then an employee information contains all its dependent instances.

So, the remaining task is to transform results into RDFS format which in our case is performed using Table 1 & 3. Whereas, Extensible Style sheet Language Transformations (XSLT) are used to transform XML into other formats in our case it is RDF. Online project X2R converter for transforming taken experiment with modifications made in the xml schema file. And the resultant RDF Schema (given is not complete, list of triples gained) are shown in Appendix Fig. A4 and A5. At last, Fig. 4 shows a directed graph of resources gained by using complete triple list.

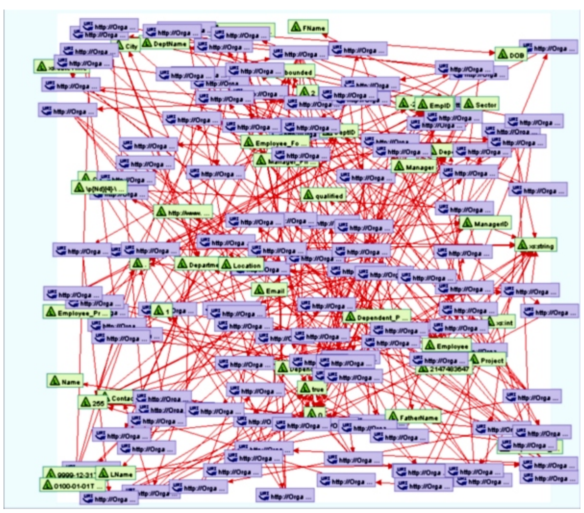

Fig. 3. "Organization" RDF Schema list translated into directed graph

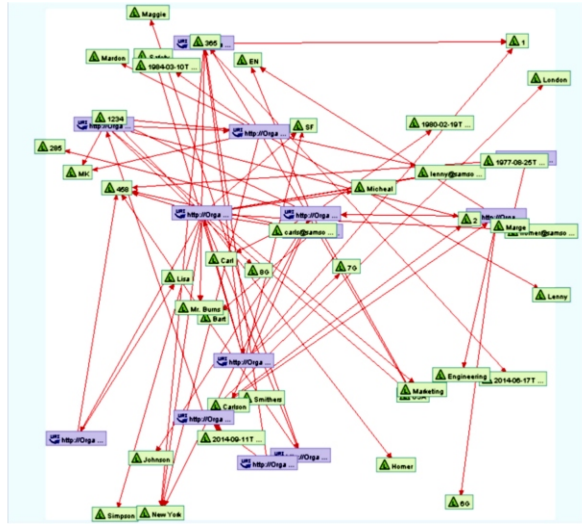

Fig. 4. "Organization" RDF triples list translated into directed graph

Then after achieving RDF file is passed it to RDF Gravity v1.0 online available tool to generate the corresponding RDF graph (shown in Fig. 3 & 4). This all have been made possible due to correct data mapping is being followed during the transformation process. Which lead us to come up with finding both schema level graphs (Fig. 3) and data level graph (Fig. 4). These graphs show us the linkage and relationship between resources used in the database of "Organization". This experiment shows that proper data type mapping is necessary to gain accurate results during the process of transformation between RDB and RDF.

C. Results

The above control experiment shows the difference between both transformations by choosing either DTD or XMLS. Whereas, DTD is less reliable than XML, as can be seen in Table IV through Fig 3.

TABLE IV
TECHNICAL COVERAGE OF DIFFERENT TOOLS USED FOR TRANSFORMATION

Technical Coverage	**Tool Capacity**				**Detail**
	RDBS	**DTD**	**XML**	**RDFS**	
Data Representation	High	Low	Average	Average	Support for data types for storage
Data Semantics	Average	Low	Average	High	Support for data based on relationships
Connector Semantics	Average	Average	Average	High	Support to relate other similar data available
Extra Functional Dependencies	High	Average	Average	Average	Support to detailed data for dependencies

In Table IV connector semantics are the triplets connecting distributed resources with having similar meanings. Distributed data connectivity and understanding becomes crucial when it comes to Web Semantics. Recourses can have different connectors with different meanings, whereas in RDB, it is not the same.

XML supports a subset of RDBS data types. As if RDFS is built upon XMLS then RDFS can support a subset of RDBS data type based superset.

Suppose X and M are Sets of data types supported by RDB and XML respectively,

Such that, M X

$$\therefore N = M \text{ ---------} \quad (1)$$

where N is Set of data types supported by RDFS

Eq. 1 shows that it is hard to transform an evolving data back and forth between RDB and RDF if data types are not properly mapped. *Then the question arises that "could XML being used as intermediate between RDB and RDF based transformation".* The answer lies in the mapping mechanism used between them which should be strong enough to support both sides to ensure equivalence of data. Experiment performed gives an evaluation for proposed mappings on the basis of equivalence of concepts.

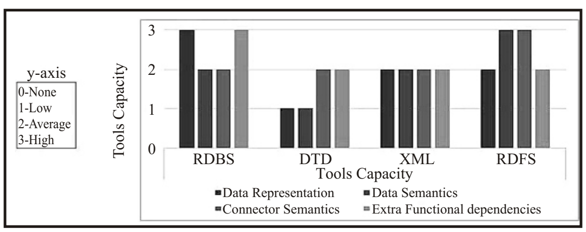

Fig. 5. Bar chart for showing Technical coverage of different tools

Results shown in Fig. 5 represents that the accuracy drops at the level of DTD compared to XMLS data type during the process of transformation .

IV. DISCUSSION

Transformation plays a key role between systems new and old to coexist. But it is not an easy task to perform. Keeping intact all pieces of information without losing accuracy is quite hard due to limitation of each mechanism used.

By using a common feature of RDF and RDB can help with enhanced utilization and compatibility among system concerning semantic web. This can assist in inducing large scale loss free transformation of traditional systems into semantically enriched systems back and forth. As the problems found, do indicate that there are many features lost during the transformation performed unidirectional or even partial bidirectional [xiii]. For now bidirectional transformation is lacking in its full capacity. It is becoming a necessity to improve compatibility among systems using either semantic or relational data models. This can be done when new improvement in the XML standard, which can hold all necessary Meta data of each table in a separate file. Through this way, this study will directly benefit data transformation process between systems containing both traditional and semantic enriched data storage.

In the Table II DTD shows biggest challenges as it only supports the data type of string. To resolve such problematic issue the transformation requires to contain the extra information related to data. On the other hand, there is a need of keeping track of metadata for bidirectional data transformation to work without any loss of information. This will make sure that data is capable to retain its original shape for either side of data models concerning semantic web and relational DB.

Whereas example section helps in understanding how XML schema can support richer concepts of database unlike DTD. By this, it is suggested that XML schema should be prepared carefully for the data transformation between RDB to RDF.

V. CONCLUSION

This research contributes to look into the weaknesses generated during transformation process in both fields of data storage and retrieval from Relational Databases (RDB) into Semantic Web. Extensive distribution of data all over the web makes accurate transformation difficult to attain. Also keeping changes intact for evolving data is hard and difficult to sustain. This leads to the need for transformation to focus more on common features of these state of the art technologies. These common features can further help in gaining better compatibility options for data transformation. By this way originality of the data can also be preserved during transformation. To achieve this, mapping helped to grasp their differences at the level of data types. It's main focus is to show common features found in both data models of RDB and RDF based schemas. Using among Document Type Definition (DTD) or Extensible Markup Language Schema (XMLS) as an intermediate which will help in improving transformation results. Then these results can be analyzed and discussed based on the given results. It is aimed to show the importance of reducing the response time of DB queries and offer compatibility between web and semantically enriched data.

VI. FUTURE WORK

Yet improvements are far from being achieved to gain the same results at both ends of systems. Using this mapping for indirect transformation can help in achieving better and enhanced results for transformation of data. Such a system can be used for retrieval and storage of data among RDB and Semantic Web with improved compatibility for bidirectional transformation. Further research on currently available tools and methodologies along with their frameworks can help for achieving state of art bidirectional transformation. It will also help us in finding weak areas and providing alternative mechanisms to transform data for bidirectional transformation.

REFERENCES

[i] O. Rusu, I. Halcu, O. Grigoriu, G. Neculoiu, V. Sandulescu, M. Marinescu, and V. Marinescu, "Converting unstructured and semi-structured data into knowledge." pp. 1-4.

[ii] G. Antoniou, and F. Van Harmelen, *A semantic web primer*: MIT press, 2004.

[iii] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel, "Semantic business process management: A vision towards using semantic web services for business process management." pp. 535-540.

[iv] P. T. T. Thuy, Y.-K. Lee, and S. Lee, "DTD2OWL: automatic transforming XML documents into OWL ontology." pp. 125-131.

[v] P. T. T. Thuy, Y.-K. Lee, S. Lee, and B.-S. Jeong, "Transforming valid XML documents into RDF via RDF schema." pp. 35-40.

[vi] P. T. T. Thuy, Y.-K. Lee, S. Lee, and B.-S. Jeong, "Exploiting XML schema for interpreting XML documents as RDF." pp. 555-558.

[vii] D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Walle, "XML to RDF conversion: a generic approach." pp. 138-144.

[viii] F. Manola, E. Miller, and B. McBride, "RDF primer," *W3C recommendation,* vol. 10, pp. 1-107, 2004.

[ix] P. Biron, A. Malhotra, and W. W. W. Consortium, "XML schema part 2: Datatypes," *World Wide Web Consortium Recommendation REC-xmlschema-2-20041028*, 2004.

[x] P. Ravindra, and K. Anyanwu, "Nesting Strategies for Enabling Nimble MapReduce Data flows for Large RDF Data," *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 10, no. 1, pp. 1-26, 2014.

[xi] M. Hert, G. Reif, and H. C. Gall, "A comparison of RDB-to-RDF mapping languages." pp. 25-32.

[xii] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat, "A survey of current approaches for mapping of relational databases to rdf," *W3C RDB2RDF Incubator Group Report*, 2009.

[xiii] D.-E. Spanos, P. Stavrou, and N. Mitrou, "Bringing relational databases into the semantic web: A survey," *Semantic Web,* vol. 3, no. 2, pp. 169-209, 2012.

[xiv] A. Shah, J. Adeniyi, and T. Al Tuwairqi, "An Algorithm for Transforming XML Documents Schema into Relational Database Schema," *Transformation of knowledge, information and data: theory and applications, Idea Group Publishing, USA*, pp. 171-189, 2005.

[xv] W. Martens, F. Neven, T. Schwentick, and G J. Bex, "Expressiveness and complexity of XML Schema," *ACM Transactions on Database Systems (TODS),* vol. 31, no. 3, pp. 770-813, 2006.

[xvi] P. Bohannon, W. Fan, M. Flaster, and P. Narayan, "Information preserving XML schema embedding." pp. 85-96.

[xvii] M. Zhou, and Y. Wu, "XML-based RDF data management for efficient query processing." p. 3.

[xviii] F. Frasincar, G.-J. Houben, R. Vdovjak, and P. Barna, "RAL: An algebra for querying RDF," *World Wide Web,* vol. 7, no. 1, pp. 83-109, 2004.

[xix] P. F. Patel-Schneider, and J. Siméon, "The Yin/Yang Web: A unified model for XML syntax and RDF semantics," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 15, no. 4, pp. 797-812, 2003.

[xx] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, "The semantic web: The roles of XML and RDF," *Internet Computing, IEEE,* vol. 4, no. 5, pp. 63-73, 2000.

[xxi] M. Grobe, "Rdf, jena, sparql and the'semantic web'." pp. 131-138.

[xxii] A. Zaveri, A. Maurino, and L.-B. Equille, "Web Data Quality: Current State and New Challenges," *International Journal on Semantic Web and Information Systems (IJSWIS),* vol. 10, no. 2, pp. 1-6, 2014.

[xxiii] D. C. Fallside, and P. Walmsley, "XML schema part 0: primer second edition," *W3C recommendation*, pp. 16, 2004.

[xxiv] J. Hendler, "Web 3.0: Chicken Farms on the Semantic Web." *IEEE Computer,* vol. 41, no. 1, pp. 106-108, 2008.

[xxv] N. Konstantinou, D.-E. Spanos, P. Stavrou and N. Mitrou, "Technically Approaching the Semantic Web Bottleneck." *International Journal of Web Engineering and Technology*, vol. 6, no. 1, pp. 83-111, 2010.

APPENDIX A

Figures taken as output in the form of XML and RDF document Samples during Experiment phase

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Department (DeptID, ManagerID, DeptName?, Sector?, Employee*)>
<!ELEMENT DeptID (#PCDATA)>
<!ELEMENT ManagerID (#PCDATA)>
<!ELEMENT DeptName (#PCDATA)>
<!ELEMENT Sector (#PCDATA)>
<!ELEMENT Dependent ANY>
<!ELEMENT EmpID (#PCDATA)>
<!ELEMENT Employee (EmpID, DeptID, FName?, LName?, FatherName?, Email?, DOB?, ContactNo?, City?,
Country?, Dependent*)>
<!ELEMENT FName (#PCDATA)>
<!ELEMENT LName (#PCDATA)>
<!ELEMENT FatherName (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT DOB (#PCDATA)>
<!ELEMENT ContactNo (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Country (#PCDATA)>
<!ELEMENT Manager (ManagerID, Location?, Project?, Name?, Department*)>
<!ELEMENT Location (#PCDATA)>
<!ELEMENT Project (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
```

Fig. A1. "Organization" RDB schema transformed into DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="Department">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="DeptID">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="2"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ManagerID">
                    <xs:simpleType>
                        <xs:restriction base="xs:int">
                            <xs:minInclusive value="-2147483648"/>
                            <xs:maxInclusive value="2147483647"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="DeptName" nillable="true" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="255"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Sector" nillable="true" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="2"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element ref="Employee" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <xs:key name="Department_PrimaryKey_1">
            <xs:selector xpath="."/>
            <xs:field xpath="DeptID"/>
        </xs:key>
        <xs:keyref name="Employee_ForeignKey_1" refer="Department_PrimaryKey_1">
            <xs:selector xpath="Employee"/>
            <xs:field xpath="DeptID"/>
        </xs:keyref>
    </xs:element>
```

Fig. A2. "Organization" RDB schema transformed into XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<OrganizationDB>
    <Manager>
        <ManagerID>1</ManagerID>
        <Location>New York</Location>
        <Project>2014-06-17T00:00:00</Project>
        <Name>Mr. Burns</Name>
        <Department>
            <DeptID>SF</DeptID>
            <ManagerID>1</ManagerID>
            <DeptName>Safety</DeptName>
            <Sector>7G</Sector>
            <Employee>
                <EmpID>458</EmpID>
                <DeptID>SF</DeptID>
                <FName>Homer</FName>
                <LName>Simpson</LName>
                <FatherName>Micheal</FatherName>
                <Email>homer@samsont.com</Email>
                <DOB>1977-08-25T00:00:00</DOB>
                <ContactNo>1234</ContactNo>
                <City>New York</City>
                <Country>USA</Country>
                <Dependent>
                    <EmpID>458</EmpID>
                    <Dependent>Marge</Dependent>
                </Dependent>
                <Dependent>
                    <EmpID>458</EmpID>
                    <Dependent>Bart</Dependent>
                </Dependent>
                <Dependent>
                    <EmpID>458</EmpID>
                    <Dependent>Lisa</Dependent>
                </Dependent>
                <Dependent>
                    <EmpID>458</EmpID>
                    <Dependent>Maggie</Dependent>
                </Dependent>
            </Employee>
        </Department>
    </Manager>
    <Manager>
        <ManagerID>2</ManagerID>
        <Location>London</Location>
        <Project>2014-09-11T00:00:00</Project>
        <Name>Smithers</Name>
        <Department>
            <DeptID>MK</DeptID>
            <ManagerID>2</ManagerID>
            <DeptName>Marketing</DeptName>
            <Sector>8G</Sector>
            <Employee>
                <EmpID>365</EmpID>
                <DeptID>MK</DeptID>
                <FName>Lenny</FName>
                <FatherName>Mardon</FatherName>
                <Email>lenny@samsont.com</Email>
                <DOB>1984-03-10T00:00:00</DOB>
                <ContactNo>1234</ContactNo>
                <City>New York</City>
                <Country>USA</Country>
            </Employee>
        </Department>
        <Department>
            <DeptID>EN</DeptID>
            <ManagerID>2</ManagerID>
            <DeptName>Engineering</DeptName>
            <Sector>6G</Sector>
            <Employee>
                <EmpID>285</EmpID>
                <DeptID>EN</DeptID>
                <FName>Carl</FName>
                <LName>Carlson</LName>
                <FatherName>Johnson</FatherName>
                <Email>carls@samsont.com</Email>
                <DOB>1980-02-19T00:00:00</DOB>
                <ContactNo>1234</ContactNo>
                <City>New York</City>
                <Country>USA</Country>
            </Employee>
        </Department>
    </Manager>
</OrganizationDB>
```

Fig. A3. "Organization" RDB transformed into XML file using XML Schema

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:OrganizationDB="http://OrganizationDB/Property/"
   xmlns:OrganizationDB1="http://OrganizationDB/Property/xmlns:">
  <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema">
   <OrganizationDB:hasResource>
    <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3">
     <OrganizationDB:hasResource>
      <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:keyref">
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:keyref/xs:field">
         <OrganizationDB:xpath>ManagerID</OrganizationDB:xpath>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:keyref/xs:selector">
         <OrganizationDB:xpath>Department</OrganizationDB:xpath>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:refer>Manager_PrimaryKey_1</OrganizationDB:refer>
       <OrganizationDB:name>Department_ForeignKey_1</OrganizationDB:name>
      </rdf:Description>
     </OrganizationDB:hasResource>
     <OrganizationDB:hasResource>
      <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:key">
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:key/xs:field">
         <OrganizationDB:xpath>ManagerID</OrganizationDB:xpath>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:key/xs:selector">
         <OrganizationDB:xpath>.</OrganizationDB:xpath>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:name>Manager_PrimaryKey_1</OrganizationDB:name>
      </rdf:Description>
     </OrganizationDB:hasResource>
     <OrganizationDB:hasResource>
      <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:complexType">
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:complexType/xs:sequence">
         <OrganizationDB:hasResource>
          <rdf:Description rdf:about="http://OrganizationDB/Resource/xs:schema/xs:element3/xs:complexType/xs:sequence/xs:element4">
           <OrganizationDB:maxOccurs>unbounded</OrganizationDB:maxOccurs>
           <OrganizationDB:minOccurs>0</OrganizationDB:minOccurs>
           <OrganizationDB:ref>Department</OrganizationDB:ref>
          </rdf:Description>
```

Fig. A4. "Manager Table of Organization DB" triples list generated using transformation algorithm XML Schema to RDFS

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:OrganizationDB="http://OrganizationDB/Property/">
  <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB">
   <OrganizationDB:hasResource>
    <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB/Manager1">
     <OrganizationDB:hasResource>
      <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB/Manager1/Department1">
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB/Manager1/Department1/Employee">
         <OrganizationDB:DeptID>EN</OrganizationDB:DeptID>
         <OrganizationDB:EmpID>285</OrganizationDB:EmpID>
         <OrganizationDB:LName>Carlson</OrganizationDB:LName>
         <OrganizationDB:FatherName>Johnson</OrganizationDB:FatherName>
         <OrganizationDB:FName>Carl</OrganizationDB:FName>
         <OrganizationDB:Country>USA</OrganizationDB:Country>
         <OrganizationDB:DOB>1980-02-19T00:00:00</OrganizationDB:DOB>
         <OrganizationDB:City>New York</OrganizationDB:City>
         <OrganizationDB:ContactNo>1234</OrganizationDB:ContactNo>
         <OrganizationDB:Email>carls@samsont.com</OrganizationDB:Email>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:Sector>6G</OrganizationDB:Sector>
       <OrganizationDB:DeptName>Engineering</OrganizationDB:DeptName>
       <OrganizationDB:ManagerID>2</OrganizationDB:ManagerID>
       <OrganizationDB:DeptID>EN</OrganizationDB:DeptID>
      </rdf:Description>
     </OrganizationDB:hasResource>
     <OrganizationDB:hasResource>
      <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB/Manager1/Department">
       <OrganizationDB:hasResource>
        <rdf:Description rdf:about="http://OrganizationDB/Resource/OrganizationDB/Manager1/Department/Employee">
         <OrganizationDB:Country>USA</OrganizationDB:Country>
         <OrganizationDB:City>New York</OrganizationDB:City>
         <OrganizationDB:ContactNo>1234</OrganizationDB:ContactNo>
         <OrganizationDB:DOB>1984-03-10T00:00:00</OrganizationDB:DOB>
         <OrganizationDB:Email>lenny@samsont.com</OrganizationDB:Email>
         <OrganizationDB:FatherName>Mardon</OrganizationDB:FatherName>
         <OrganizationDB:FName>Lenny</OrganizationDB:FName>
         <OrganizationDB:DeptID>MK</OrganizationDB:DeptID>
         <OrganizationDB:EmpID>365</OrganizationDB:EmpID>
        </rdf:Description>
       </OrganizationDB:hasResource>
       <OrganizationDB:Sector>8G</OrganizationDB:Sector>
       <OrganizationDB:DeptName>Marketing</OrganizationDB:DeptName>
       <OrganizationDB:ManagerID>2</OrganizationDB:ManagerID>
       <OrganizationDB:DeptID>MK</OrganizationDB:DeptID>
      </rdf:Description>
     </OrganizationDB:hasResource>
     <OrganizationDB:Name>Smithers</OrganizationDB:Name>
     <OrganizationDB:Project>2014-09-11T00:00:00</OrganizationDB:Project>
     <OrganizationDB:Location>London</OrganizationDB:Location>
     <OrganizationDB:ManagerID>2</OrganizationDB:ManagerID>
    </rdf:Description>
   </OrganizationDB:hasResource>
```

Fig. A5. "Organization DB" triples list generated using transformation algorithm XML data file to RDF