Semantic RDF Based Integration Framework for Heterogeneous XML Data Sources

Deniz KILINÇ[#1], Fatma BOZYİĞİT[#2], Pelin YILDIRIM[#3]

[#] *Department of Software Engineering, Celal Bayar University, Turkey*

***Abstract*— A significant amount of data on the Web is in the XML format or may easily be converted to XML or to its variations. XML is still the most appropriate language for data interchange and serialization. In this paper, a new framework which can integrate any heterogeneous XML data sources is presented. Each data source is translated into semantically meaningful regular expressions without changing original data source. Proposed framework has two major phases for data preparation. In the first phase, each data source is processed to obtain regular expressions which accommodate with the design choices that made in target by utilizing known global semantic vocabulary as an input. The second phase combines these regular expressions to get a global schema by preserving the original source data. A regular expression generator tool which produces regular expressions by regarding vocabulary and an integrator tool box which integrates and processes regular expressions, are also introduced.**

***Keywords*— Integration, XML Data Source, RDF, XPath, XQuery.**

I. INTRODUCTION

In real life scenarios, varied database servers such as Microsoft SQL Server, Oracle and DB2-400 can store and manipulate data. When data is spread across various databases, there will be some integration needed to be performed. On the other hand, information on the Web that is overwhelmingly in HTML, XML, RDF formats is rapidly growing day by day. One of the most important problems encountered in web application integration is heterogeneity of the distributed data sources. Consequently, a common standard is required for data interchange and serialization. World Wide Web Consortium (W3C) announced Extensible Markup Language (XML) [1], [2] technology as a universal data exchange format between applications and organizations. While XML tags describe the structure and semantics of a document's content, they do not define the format of the data like HTML. XML is easy to learn, implement, read, and test. XML is an effective, portable and easily customized data format that can easily sent over through any protocol. As a point of Business-to Business (B2B) [13] view, XML supports shortened product development time for most XML-related data exchange. If data is exchanged with a partner in B2B, some form of Electronic Data Interchange (EDI) [12], which is costly, will be probably used by users.

Another requirement which is XML integration arises with respect to above situation. Difficulty of integration depends on the degree of XML data source homogeneity. If data that is intended for integration is homogenous, there will no need to data transformation, because of the matching consistency among document schemas. But, in real world, different XML source elements and attributes may have different names, element orders and data types in their schemas, due to the modification needs in requirements. Therefore, real world integrations will not be as simple as like in homogenous data integration.

To solve this integration problem, semantic RDF (Resource Description Framework) [3] based regular expressions can be used. In this paper, regular expressions are utilized to define the whole structure semantically, instead of using XPath (XML Path Language) [4] or XSLT (Extensible Style sheet Language) [5] technologies. Methodologies of RDF's resource definition express how the final result of our work, which is "regular expressions", should be represented. Proposed solution is easier and more effective compared to other works in use. Because a novel method aims to construct logical vocabulary instead of interfering XML documents directly.

The rest of the paper is organized as follows: Section 2 gives a summary of literature review and information about other studies which tries to perform integration of different XML data sources. In Section 3, detailed explanation of proposed work and solution to XML integration problem are described. Section 4 explains and implementation of proposed solution over a real life scenario. Finally, Section 5 concludes the paper and gives a look at the future studies on this subject.

II. RELATED WORKS

Researchers try to solve integration problem, using a predefined global schema [6], [7] for the heterogeneous XML data sources which have their own local schemas. Their solutions require much complicated rules for mapping from local schemas to a global schema. So A. Halevy et al., [6] and L. Popa et al., [7] are both unsuccessful in some points. L. Popa et al., [7], while realizing mapping process and after then converting them to queries, exchange operation of unstructured data or documents are not regarded. A. Halevy et al., [6] presents a solution which has a complex structure because the system has three parts and results of these three parts are evaluated separately. Therefore, this situation makes their study more complicated.

Mappings are really too complex for IT administrators because they have to know XPath and XSLT in detail. Mappings may not be one-to-one match from local schema to global-one, because an element in local schema may not exist in global-one, vice-versa. A. Halevy et al., [6] and L. Popa et al., [7] solve this problem by adding empty XML elements into local data sources that changes the original structure. Hence too much redundant storage space is spent.

B. Amann et al., [8] suggests general solutions and mechanisms for realizing the goals of Semantic-Web. They indicate that manipulation of XML data to understand their expressions is necessary to enable developing efficient applications for real world. The aim of their study is to create a mediator that is used on XML for data integration. They present a prototype as a local view and give the users a virtual data repository in a given domain. Thus, after collecting the necessary data, they put them in virtual local repositories from the external sources that are independent from the each other. In this study OQL is chosen as query language and XQuery as semi-structured language. Their method has disadvantages in terms of complexity because of having so many expressions are used while mapping between OQL and XQuery. This situation makes the translation algorithm more complex.

Researchers explain a new RDF Based architecture which has five layers for integration of heterogeneous information sources in their study [9]. They create an architecture that includes a global RDF mediator to collect data from heterogeneous XML data sources. This study also combines semantic and on-demand driven retrieval. In demand retrieval, data is collected dynamically from integrated sources. Researchers also prepare a semantic interface which is implemented for reaching to heterogeneous information sources.

I. F. Cruz et al., [10] presents a new approach which uses layered system to solve data integration problem. They focus on semantic layer to create solution of data interoperability for construction of local schemas and they benefit from RDF. Data is transformed to RDF files and shared in RSSDB (RDF Schema Specific Database). Their proposed system has a dictionary which gives relationship between ontology and each schema.

Differently from previous studies, applicability of our proposed study has crucial importance in terms of creating new applications, which have different approaches. For example, existing applications such as BizTalk, IBM Websphere and Software AG WebMethods can be redesigned by exploiting suggestions which are in background of proposed solution.

III. Description of Method

Since XML is an open data model and cannot do any integration automatically by itself, data integration with XML is an ongoing problem. Eventually, a third party solution must be required to solve it. Solutions mentioned in section 2, cause new problems during integration of XML data sources. In this paper, a new approach to solve these problems is suggested.

Our approach recommends a semantic RDF-based perspective to XML integration problem. Notion of semantics in proposed study is mapping documents to a target format from any data source and is obtaining regular expressions. RDF is acronym of Resource Description Framework [3] that is a W3C-recommended XML application to encode and exchange, reuse structured metadata, define and give a meaning to resources on World Wide Web. An RDF document abstracts the details of document by presenting a semantic layer. The following RDF document describes an example web resource which has three parts. The first one is an identifier which can be URI, URL or ISBN, the second is a property, “Author” in the example and an object which is the value of property, “Sibel KILINÇ” in the example as shown in Fig. 1.

```
<rdf: RDF xmlns:rdf="http://www.w3.org/1999/ 02/22-rdf-syntax-ns#">
  <rdf: Description about="http://www.deu.edu.tr">
    <Author>Sibel KILINÇ</Author>
  </rdf: Description>
</rdf: RDF>
```

Fig. 1 RDF document example.

RDF-based regular expressions are utilized to define local XML sources. These expressions both abstract the business logic and simplify the difficulties of local XML schemas by defining an id, a property and property’s value. The following regular expression is an output of the tool proposed in the rest of paper and defines a predicate rule which is an element of logical vocabulary and named “STOCK-NAME” in Fig. 2.

```
STOCK-NAME(xS,xN) := Source/STOCK xT, xT/ID xS, xT/NAME xN
```

Fig. 2 RDF-based regular expressions.

In this predicate rule, Source is the name of local schema. Each STOCK element of Source is assigned to a temporal variable, xT. Each xT has two properties named ID and NAME. xS is an identifier and holds the ID of STOCK. xN is the name of property and value of xN holds NAME of STOCK.

Semantic RDF-based Integration Framework has two major tools:

1. Regular Expression Generator Tool (REGT)
2. Integrator Tool Box (ITB)

Suppose that N is the number of local XML data sources that will be integrated. These data sources are also input for the Regular Expression Generator Tool (REGT). The other input for REGT is Global Semantic Vocabulary, which consists of common XML elements in all local XML data sources with an identifier and a property as in RDF. The left part of “STOCK-NAME” description is an example element of vocabulary. REGT produces N regular expression sets with respect to local data sources. The right part “STOCK-NAME” description is an example output of REGT. The upper part of framework presents the processes done at local.

Integrator Tool Box (ITB) which is shown in Fig. 3 combines right sides of the regular expression with the guide of vocabulary elements using some iteration expressions. Finally, Global XML Data source is generated without changing the structure of local XML data sources, unlike related works.

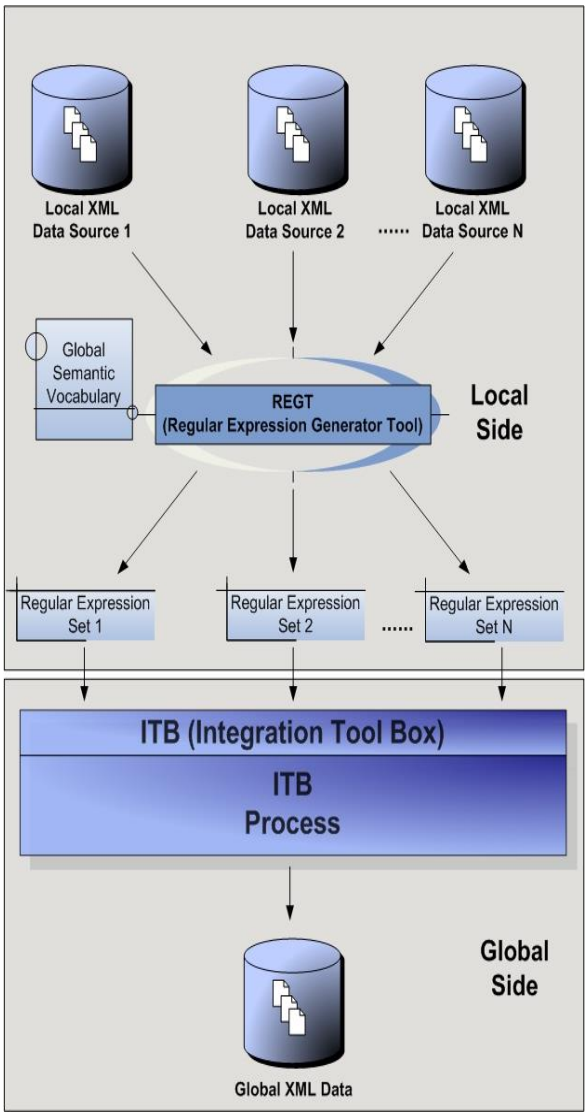

Fig. 3 ITB architecture overview.

The following section describes how integration process is realized by presenting a real life scenario.

IV. A Real Life Scenario

Suppose that there is a corporation which has a branch office located in İzmir and has a head office located in Ankara. Each office holds different XML schemas for daily orders. Manager of the corporation wants to see some audit reports, so that all orders must be consolidated. At this point, an integration problem occurs and a solution is needed.

General solution to this integration problem consists of the following steps:

1. A global definition and a global schema must be agreed on.
2. For each data source, different mappings must be applied to transform local schemas into global one schema.
3. All transformed data sources must be consolidated for querying.

We may solve this integration need using the today's standard XML technologies with following steps:

1. A global XML Schema is defined.
2. XPath or Xquery [11] mappings are used for element transformation which changes the structure of original local schemas.
3. Complex and redundant XSL-Transformations are used for joining step of transformation.

Although integration problem is able to solve with these three steps, new problems reveal. Firstly, matching problem will occur, when XML elements in local data sources do not exist in global one. A. Halevy et al., [6] and L. Popa et al., [7] solve this problem by adding empty XML elements into local data sources that changes the original structure. As a result, too much redundant storage space is spent because of this solution. Secondly, to apply transformation for joining steps, IT administrators of each branch must know complex XPath mappings and XSLT transformations.

In our proposed framework, the encountered problems are resolved with the following steps:

1. A common vocabulary is agreed on.
2. Instead of adding new elements to schema structures, generating RDF-based regular expressions from local data sources by using REGT which facilitates the transformation process of integration rather than formulating the complex Xpath queries.
3. Regular expressions are integrated and processed by using ITB that realizes the joining process. Thanks to ITB and REGT, IT administrators do not need to know XSLT and Xpath technology.

A. The Design of Solution Using Proposed Approach

Common vocabularies and predicate rules are specified in the first step. Applied predicate rules to this scenario are listed below:

- STOCK-ORDER(xS,xO),
- STOCK-NAME(xS,xN),
- STOCK-QUANTITY(xS,xQ),
- ORDER-DATE(xO,xD)
- ORDER-CUST(xO,xC)

These predicate rules are the elements of common vocabulary which is also known as global definition.

Table 1 and Table 2 present the local schema information and a sample XML document of head office and branch office.

TABLE I
LOCAL SCHEMA AND A SAMPLE XML DOCUMENT OF HEAD OFFICE, ANKARA

```
<schema>
 <element name=”ORDERLIST”>
 <complexType>
  <sequence>
   <element name=”ORDER”
     type=”typeOrder”/>
  </sequence>
 </complexType>
 <complexType
   name="typeOrder">
  <element name="NUMBER"
    type="string"/>
  <element name="DATE"
    type="DateTime"/>
   <element name="CUSTID"
    type="string"/>
   <element name="STOCK"
    type="typeStock"/>
 </complexType>
 <complexType
    name="typeStock">
   <element name="ID"
    type="string"/>
   <element name="NAME"
    type="string"/>
   <element name="QUANTITY"
    type="integer"/>
 </complexType>
 </element>
</schema>
```

```
<?xml version="1.0"?>
<ORDERLIST>
 <ORDER>
  <NUMBER>AN001</NUMBER>
<DATE>01/03/2004</DATE>
<CUSTID>0001</CUSTID>
<STOCK>
 <ID>S0001</ID>
 <NAME>CANON</NAME>
 <QUANTITY>
   3
 </QUANTITY>
</STOCK>
 </ORDER>
 <ORDER>
  <NUMBER>AN002</NUMBER>
<DATE>01/03/2004</DATE>
<CUSTID>0002</CUSTID>
<STOCK>
 <ID>S0005</ID>
  <NAME>PANASONIC</NAME>
 <QUANTITY>
   10
 </QUANTITY>
</STOCK>
 </ORDER>
</ORDERLIST>
```

In Table 1, REGT tool takes common semantic vocabulary and a local schema of Ankara as inputs and generates the following regular expression set 1 with respect to the request of manager.

STOCK_ORDER regular expression lists the stocks with ID and orders with NUMBER as shown in Fig. 4. This regular expression provides to get all information about orders and related stocks.

STOCK-ORDER(xS,xO) ::= Ankara/ORDERLIST/ORDER xT, xT/STOCK/ID xS, xT/NUMBER xO

STOCK-NAME(xS,xN) ::= Ankara/ORDERLIST/ORDER/STOCK xT, xT/ID xS, xT/NAME xN

STOCK-QUANTITY(xS,xQ) ::= Ankara/ORDERLIST/ORDER/STOCK xT, xT/ID xS, xT/QUANTITY xQ

ORDER-DATE (xO,xD)::= Ankara/ORDERLIST/ORDER xT, xT/NUMBER xO, xT/DATE xD

ORDER-CUST(xO,xC)::= Ankara/ORDERLIST/ORDER xT, xT/NUMBER xO, xT/CUSTID xC

Fig. 4 Regular expression set generated for Ankara.

TABLE II
LOCAL SCHEMA AND A SAMPLE XML DOCUMENT OF BRANCH OFFICE, IZMIR

```
<schema>
 <element name="ORDLIST">
<complexType>
 <sequence>
   <element name="ORD" type="typeOrd"/>
 </sequence>
</complexType>
<complexType name="typeOrd">
 <element name="ORDID" type="string"/>
 <element name="CUSTID" type="string"/>
 <element name="DATE" type="DateTime"/>
 <element name="STKLIST"
   type="typeStList"/>
</complexType>
<complexType name="typeStList">
 <element name="STK" type="typeSt"/>
  </complexType>
<complexType name="typeSt">
 <element name="ID" type="string"/>
 <element name="QUANT" type="integer"/>
 <element name="NAME" type="string"/>
</complexType>
 </element>
</schema>
```

```
<?xml version="1.0"?>
<ORDLIST>
 <ORD>
  <ORDID>IZ001</ORDID>
  <CUSTID>0005</CUSTID>
  <DATE>01/03/2004
  </DATE>
 <STKLIST>
   <STK>
<ID>S0006</ID>
<QUANT>2</QUANT>
    <NAME>
        NIKON</NAME>
   </STK>
  </STKLIST>
 </ORD>
 <ORD>
<ORDID>IZ005</ORDID>
<CUSTID>0008</CUSTID>
<DATE>01/03/2004</DAT>
<STKLIST>
  <STK>
   <ID>S0005</ID>
   <QUANT>5</QUANT>
   <NAME>
    PANASONIC
   </NAME>
  </STK>
 </STKLIST>
</ORD>
</ORDERLIST>
```

In Table 2, REGT produces the following regular expression set 2.

STOCK-ORDER(xS,xO) ::= İzmir/ORDLIST/ORD xT, xT/STKLIST/STK/ID xS, xT/ORDID xO

STOCK-NAME(xS,xN) ::= İzmir /ORDLIST/ORD/STKLIST/STK xT, xT/ID xS, xT/NAME xN

STOCK-QUANTITY(xS,xQ) ::= İzmir/ORDLIST/ORD/STKLIST/STK xT, xT/ID xS, xT/QUANT xQ

ORDER-DATE(xO,xD)::= İzmir /ORDLIST/ORD xT, xT /ORDID xO, xT/DATE xD

ORDER-CUST(xO,xC)::= İzmir /ORDLIST/ORD xT, xT/ORDID xO, xT/CUSTID xC

Fig. 5 Regular expression set generated for İzmir.

Regular expression set 1 and set 2 are prepared for consolidation purpose and they are processed by ITB. At his point, if we assume a scenario in which a manager wants to report the orders for 'PANASONIC' stock in each office, then the following code script in Table 3 is generated and is run by using ITB.

TABLE III
OUTPUT OF ITB

```
OutXML(<ORDERLIST>);
For $xO in each
  (STOCK-ORDER(xS,xO))
  If ($xS = STOCK-
NAME(xS,xN))
 And
 ($xN = 'PANASONIC')
 And
 ($xS = STOCK-
     QUANTITY(xS,xQ))
 Then
OutXML(
   <ORDEREDSTOCK>
<ORDID>$xO</ORDID>
<NAME>$xN</NAME>
<QUANT>$xQ</QUANT>
    </ORDEREDSTOCK>
);
   End If
Next
OutXML(</ORDERLIST>);
```

```
<ORDERLIST>
 <ORDEREDSTOCK>
 <ORDID>AN002</ORDID>
<NAME>
  PANASONIC
</NAME>
<QUANT>10</QUANT>
 </ORDEREDSTOCK>
 <ORDEREDSTOCK>
 <ORDID>IZ005</ORDID>
<NAME>
  PANASONIC
</NAME>
<QUANT>5</QUANT>
 </ORDEREDSTOCK>
</ORDERLIST>
```

OutXML is a special function to generate XML elements. The "if" condition checks the given condition and if it is true, it executes the statement in the following first line. The "for-each" traverses STOCK-ORDER regular expression and as part of the loop a variable named $xO is created that stock orders. Finally, an XML document is generated by including the elements; order id (ORDID), stock name (NAME), and stock quantity (QUANT) for reporting.

V. CONCLUSIONS

XML is still the most appropriate language for data interchange and serialization. In this paper, a new framework which can integrate any heterogeneous XML data sources is presented. New tools for schema transformation and regular expression integration are also suggested. Proposed study differs from related works in terms of advantages.

This system guarantees that the global data source includes all the elements and attributes which have defined as a parameter in global semantic vocabulary. Instead of adding new elements to schema structures, RDF-based regular expressions from local data sources are generated by using REGT which facilitates the transformation process of integration rather than formulating the complex Xpath queries. This vocabulary can also be changed manually according to requests of corporation.

Currently suggested tools REGT and ITB may quickly generate and integrate regular expression sets from different data sources. Eventually, while other studies may cause excessive memory usage and disk usage, this study gets close to optimal solution.

REFERENCES

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, Markup Language(XML) 1.0 W3C Recommendation, February 1988.

[2] D. Kılınç and A. Kut, "XML teknolojisine gerçekçi yaklaşım", in *Türkiye'de Internet" Konferansı (Inet-tr'09),* 2003.

[3] (2014) RDF specification. [Online]. Available: http:// www.w3c.org/RDF/

[4] (1999) XPath specification. [Online]. Available: http://www.w3.org/TR/xpath/.

[5] J. Clarke, XSL Transformations (XSLT) version 1.0. W3C Recommendation, November 1999.

[6] A. Halevy, O. Etzioni, A. Doan, Z. Ives, J. Madhavan, L. McDowell and I. Tatarinov, Crossing the Structure Chasm, in *CIDT'03,* 2003.

[7] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez and R. Fagin, "Translating Web Data", in *Proceedings of the 28th VLDB Conference,* pp. 598–609, 2002.

[8] B. Amann, C. Beeri, I. Fundulaki and M. Scholl, "Ontology-based integration of XML web resources", *The Semantic Web — ISWC 2002,* vol. 2342, pp. 117–131, May 2002.

[9] R. Vdovjak and G. Houben, "RDF Based Architecture for Semantic Integration of Heterogeneous Information Sources", in *International Workshop on Information Integration on the Web*, pp. 51-57, April 2001.

[10] I. F. Cruz, H. Xiao, and F. Hsu. "An Ontology-based Framework for Semantic Interoperability between XML Sources", *in Eighth International Database Engineering & Applications Symposium (IDEAS 2004)*, July 2004.

[11] (2004) XQuery specification [Online]. Available: http://www.w3.org/XML/Schema.

[12] G. Premkumar, K. Ramamurthy and Sree Nilakanta "Implementation of Electronic Data Interchange: An Innovation Diffusion Perspective", *Journal of Management Information Systems*, vol. 11, no. 2, pp. 157-186, Fall 1994.

[13] P. Godefroid, and W. Pförtsch, "Business-to-business-marketing", Kiehl, 2003.