ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Detection of Repetition Data in XML Storage

Gajanan Temgire, Vinod Wadne

Master of Engineering, Department of Computer Engineering, Dhole Patil College of Engineering, Wagholi, Pune, India

Professor, Department of Computer Engineering and Technology, JSPMs Imperial College of Engineering & Research, Wagholi, Pune, India

ABSTRACT: Data in any system is core part of which day by day increasing in size. There are heterogeneous programming languages, programming models and architectures which communicate each other to exchange data among selves. To fulfill the exchange of data across the system standard used is XML data. As there is not any standard schema and validation strategy across the system there is increase in duplicate data between the diffrrent systems. That's why duplicate data detectionbecomes an important optimization criterion in XML bases interoperable systems. This paper attempts to provide ahierarchical tree based algorithm to detect the differentiation between different data versions. The proposed algorithm finds the hash key and position of the each node in XML to compare data. This paper covers the homogeneous XML file comparison and heterogeneous data comparisons. Heterogeneous data comparison involves comparison between relational database and XML based database.

KEYWORDS: XML, XML Differentiation, optimization, differentiation, Heterogeneousdata.

I. INTRODUCTION

As software world evolve, the electronic data usage, importance and security increasedlargely. As data extent and large data usage the capacity and storage of data plays animportant role in representation and selection of data. There are many ways to represent andoptimization of data. Relational database uses the normalization to optimizethe data. Priority task is to deal data storage which has redundant storage. There is none of the beneficiaryto have the duplicate and redundant data. There are not key concepts of keys (like primarykey, foreign key etc.) for Hierarchical database (XML) and that why it's a difficult to identifythe similarity and duplication in XML. This leads to think in such direction which identifiesthe duplication data in the Hierarchical database. As popular and inter operable universalhierarchical data format is XML, our research revolves around the XML data duplication.

We present here probabilistic duplicate detection algorithm for Hierarchical data calledas DifferentiationXML. This algorithm considers both similarity of attribute contents andthe relative importance of descendent elements with respect to overall similarity score. We propose the hierarchical tree based algorithm for duplicate detection. We first construct tree structure model for duplicate detection and then show how this model is used to computethe similarity between XML object representations. Given this similarity, we classify two XML objects as duplicate by task of identifying and discovering semantic relationship between elements of two or moreschemas. It plays important roles for many database applications, such as data integrationto identify and characterize inter-schema relationships between multiple (heterogeneous)schemas, data warehousing to map data sources to a warehouse schema, E-business tohelp map messages between different XML formats. Heterogeneous data matching andduplicate detection between relational data and hierarchical data is also vital part in futureduplication detection of heterogeneous data.

Relational data is the structured row columns formatted data which is easy to project and normalize. Relational database like SQL Server, Oracle has the keys like primary key, foreign key, unique and identity key are used to prevent inserting redundant data into database. Our research interest and aim is of the hierarchical data system eXtensible Markup Language (XML) as it is interoperable data format which is universally accepted data for storage and data transfer. In XML we have hierarchical data using nodes, elements and attributes. Node has combination of values and child nodes. Child nodes again can have multiple values and again its child nodes [2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

II. RELATED WORK

Efficient and Effective Duplicate Detection in Hierarchical Data focuses on the homogeneous xml schema duplicate detection using the Bayesian network. XMLDup is the algorithm which used the Bayesian network and probability model to find out the duplicate in XML data. XMLDup also proposed the network pruning for the XMLDup algorithm for efficient and effective detection of repetition of data [1]. It plays important roles for many database applications, such as data integration to identify and characterize inter-schema relationships between multiple (heterogeneous) schemas, data warehousing to map data sources to a warehouse schema, E-business to help map messages between different XML formats. Heterogeneous data matching and duplicate detection between relational data and hierarchical data is also vital part in future duplication detection of heterogeneous data. We proposed the XMLDiff algorithm as base for the heterogeneousXML schema differentiation identification. Main challenge and going forward of research is to find out the differentiation between relational database and hierarchical database [3]. After years of research on ontology matching, it is reasonable to consider several questions: is the field of ontology matching still making progress? Is this progress significant enough to pursue further research? If so, what are the particularly promising directions? It conjectures that significant improvements can be obtained only by addressing important challenges for ontology matching. We present such challenges with insights on how to approach them, thereby aiming to direct research into the most promising tracks and to facilitate the progress of the field [4]. Among the studies of duplication between hierarchical and relational database the work [5] focuses on to convert hierarchical XML data into table format of the relational database. Some studies [6] surveyed the XML duplicate record detection which covers the field matching techniques and duplicate record detection techniques. There was the character based similarity metric and token based similarity metrics explanation for the filed matching techniques. Schema matching is the problem of generating correspondences between elements of two schemas. A schema is a formal structure that represents an engineered artifact, such as a SQL schema, XML schema, entity-relationship diagram, ontology description, interface definition, or form definition. A correspondence is a relationship between one or more elements of one schema and one or more elements of the other. There are many applications that require schema matching. In the database field, it is usually the first step in generating a program or view definition that maps instances of one schema into instances of another. For example, it arises in object-to-relational mappings; data warehouse loading, data exchange, and mediated schemas for data integration [7].

III. PROPOSED ARCHITECTURE

Proposed work has three different modules which will be presented here. We willhave the three modules like Homogeneous hierarchical database schema structure, Heterogeneoushierarchical database schema structure, Heterogeneous database (Hierarchicaland Relational database) schema. Below subsections depicts the system architecture of themodules. Hierarchical database in the research project will be eXtensible Markup Language(XML). Below subsections depicts the system architecture of the modules

A. Homogeneous hierarchical database schema structure: XML Schema will have the homogeneous structure. XML schema in this module is going to be the identical schema and identical structure of the XML elements, attributes and sequence.

B. Heterogeneous hierarchical database schema structures: XML Schema will have the heterogeneous i.e. different structure. XML schema in this module is going to be the different schema and different structure of the XML elements, attributes and sequence.

C. Heterogeneous databases (Hierarchical and Relational database) schema In this module of research the duplicate detection would be between XML (hierarchical database) and relational database. There will be mapping mechanism by which the table structure and XML schema structure need to be matched.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

IV. PROPOSED ALGORITHM

Change detection algorithms are used to find the differences between two documents and to construct a delta representation of these differences. It can be used for archiving systems, versioning, or just for a small requirement in your general application. We have two XML files and we will generate a delta file to save the changes. There is no standard for delta files.

A. *Initialization:*

In this step, a hash value for each node is generated andadded to each node with the ID attribute. The POS attributeis also added. We need the ID and POS attributes to be able to make joins in XML.

POS = Parent nodes ancestors count +"-" + Parent nodesright siblings count + Nodes ancestors count +"-" + Nodesright siblings count. Figure 5 shows the tree with each nodesPOS.

B. *Detecting Changes:*

In First step our XML tree with POS values is ready. In this step we will do the join operations on the XML nodes. XML join operation will give us the identical elements having same POS and data. Depend on the POS information the resultant XML will be created. Resultant XML will be created by using move, delete and insert operations on XML nodes.

V. RESULTS

Let's take the example to calculate the delta XML file for XML element differentiation.To calculate the delta we need to take first the two XML data sets/files. There will bedifferentiation calculation between versions 1 with version 2. As shown in figure 1 firstversion of XML file is presented. POS calculation on the first version of XML file is shownin figure 3.As shown in figure 2 second version of XML file is presented. POS calculationon the second version of XML file is shown in figure 4.After the XML LINQ Diff algorithm on the given input XML files the resulted XML deltafile is shown in figure 5.

```
<employees>
  <employee salaried="yes">
    <name>TEST DELETED</name>
    <hire_date>2/6/1998</hire_date>
  </employee>
  <employee salaried="no">
    <name>Gustavo Achong</name>
    <hire_date>7/31/1996</hire_date>
  </employee>
  <employee salaried="no">
    <name>Flunk</name>
    <hire_date>7/5/1996</hire_date>
  </employee>
  <employee salaried="no">
    <name>Six seven times</name>
    <hire_date>8/31/1996</hire_date>
  </employee>
</employees>
```

```
<employees>
  <employee salaried="no">
    <name>Gustavo Achong</name>
    <hire_date>7/31/1996</hire_date>
  </employee>
  <employee salaried="no">
    <name>Flunk</name>
    <hire_date>7/5/1996</hire_date>
  </employee>
  <employee salaried="no">
    <name>Six seven times</name>
    <hire_date>8/31/1996</hire_date>
  </employee>
  <employee salaried="yes">
    <name>ZEYNEP - Inserted</name>
    <hire_date>1/1/2009</hire_date>
  </employee>
</employees>
```

Fig.1.Input XML File 1(Version1.xml)Fig. 2. Input XML File 2(Version2.xml)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

```
<employees>
  <employee salaried="yes" ID="F1-1E-1E-92-8F-40-80-BE-53-C7-D9-7E-C8-5D-D9-62" POS="0-0-1-3">
    <name ID="9E-61-94-09-C7-40-11-AB-8D-69-0D-DA-D7-D5-C7-7C" POS="1-3-2-1">TEST DELETED</name>
    <hire_date ID="6B-C3-7A-51-C2-5F-46-EA-51-21-81-A1-43-C5-D1-F1" POS="1-3-2-
0">2/6/1998</hire_date>
  </employee>
  <employee salaried="no" ID="F6-53-3D-9C-28-43-5B-B7-E7-B4-D9-A9-FE-03-97-FE" POS="0-0-1-2">
    <name ID="D3-F0-D5-5B-9D-C6-0B-8A-EB-27-67-B5-AD-2F-66-0E" POS="1-2-2-1">Gustavo Achong</name>
    <hire_date ID="D2-34-63-05-86-49-DB-BF-8E-AC-CE-83-8A-47-5D-38" POS="1-2-2-
0">7/31/1996</hire_date>
  </employee>
  <employee salaried="no" ID="D8-5D-1D-20-2F-8B-3B-F9-5E-CF-03-D7-B0-81-83-69" POS="0-0-1-1">
    <name ID="A9-2F-B7-89-CE-47-01-B8-2A-95-87-11-A8-0D-07-EF" POS="1-1-2-1">Flunk</name>
    <hire_date ID="34-70-7D-A1-F1-70-52-6D-B8-40-65-86-89-2D-22-67" POS="1-1-2-
0">7/5/1996</hire_date>
  </employee>
  <employee salaried="no" ID="61-B7-E4-72-12-E4-63-53-A8-E4-E4-E5-60-0E-EE-75" POS="0-0-1-0">
    <name ID="3D-9D-B7-D4-F6-62-F7-55-12-49-3C-E6-DA-89-3A-8F" POS="1-0-2-1">Six seven times</name>
    <hire_date ID="EC-71-F3-93-86-56-3D-F1-34-D5-1E-1D-09-50-95-D6" POS="1-0-2-
0">8/31/1996</hire_date>
  </employee>
</employees>
```

Fig. 3.XML File 1 (Version1.xml) after hash and POS Algorithm

```
<employees>
  <employee salaried="no" ID="F6-53-3D-9C-28-43-5B-B7-E7-B4-D9-A9-FE-03-97-FE" POS="0-0-1-3">
    <name ID="D3-F0-D5-5B-9D-C6-0B-8A-EB-27-67-B5-AD-2F-66-0E" POS="1-3-2-1">Gustavo Achong</name>
    <hire_date ID="D2-34-63-05-86-49-DB-BF-8E-AC-CE-83-8A-47-5D-38" POS="1-3-2-
0">7/31/1996</hire_date>
  </employee>
  <employee salaried="no" ID="D8-5D-1D-20-2F-8B-3B-F9-5E-CF-03-D7-B0-81-83-69" POS="0-0-1-2">
    <name ID="A9-2F-B7-89-CE-47-01-B8-2A-95-87-11-A8-0D-07-EF" POS="1-2-2-1">Flunk</name>
    <hire_date ID="34-70-7D-A1-F1-70-52-6D-B8-40-65-86-89-2D-22-67" POS="1-2-2-
0">7/5/1996</hire_date>
  </employee>
  <employee salaried="no" ID="61-B7-E4-72-12-E4-63-53-A8-E4-E4-E5-60-0E-EE-75" POS="0-0-1-1">
    <name ID="3D-9D-B7-D4-F6-62-F7-55-12-49-3C-E6-DA-89-3A-8F" POS="1-1-2-1">Six seven times</name>
    <hire_date ID="EC-71-F3-93-86-56-3D-F1-34-D5-1E-1D-09-50-95-D6" POS="1-1-2-
0">8/31/1996</hire_date>
  </employee>
  <employee salaried="yes" ID="F9-88-00-00-FD-3A-9F-B4-9B-87-9B-DA-7B-02-2F-82" POS="0-0-1-0">
    <name ID="FC-42-76-55-BF-0F-83-CB-27-C2-0A-BD-E2-3F-FC-9C" POS="1-0-2-1">ZEYNEP - Inserted</name>
    <hire_date ID="C0-16-61-D5-08-71-02-26-71-9C-A2-F0-E6-68-CD-E8" POS="1-0-2-
0">1/1/2009</hire_date>
  </employee>
</employees>
```

Fig.4.XML File 2(Version2.xml) after hash and POS Algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

```
<?xml version="1.0" encoding="UTF-8"?>
- <Delta To="Version2.xml" From="Version1.xml">
  - <Delete>
    - <employee POS="0-0-1-3" salaried="yes">
        <name POS="1-3-2-1">TEST DELETED</name>
        <hire_date POS="1-3-2-0">2/6/1998</hire_date>
      </employee>
  </Delete>
  - <Insert>
    - <employee POS="0-0-1-0" salaried="yes">
        <name POS="1-0-2-1">ZEYNEP - Inserted</name>
        <hire_date POS="1-0-2-0">1/1/2009</hire_date>
      </employee>
  </Insert>
  - <Move>
      <employee To="0-0-1-3" From="0-0-1-2"/>
      <employee To="0-0-1-2" From="0-0-1-1"/>
      <employee To="0-0-1-1" From="0-0-1-0"/>
  </Move>
</Delta>
```

Fig. 5.Resulted XML file with delta

VI. CONCLUSION AND FUTURE WORK

We concentrated on the heterogeneous XML schema differentiation. It will be veryuseful to have differentiation to compare the XML data across the different organizations.We are also differentiating the data between heterogeneous database i.e. differentiationbetween relational database and hierarchical database. The heterogeneous database duplicationsystem is the new forthcoming feature in a duplicate detection system.There are many stages where manual intervention is required in the proposed research.Future scope on top of the proposed would be the removal of manual intervention in theduplicate detection. Mapping, probability selection required the manual intervention whichcan be future work for the proposed research. Various concepts similar to relation databasecan be applied in the XML data. Concepts like normalization, query tuning, and queryoptimization can be introduced to XML data.

REFERENCES

1. Luis Leitao, PavelCalado, and Melanie Herschel, 'An Efficient and Effective Duplicate Detection in Hierarchical Data', IEEE Transactions On Knowledge And Data Engineering, Vol. 25, Issue. 5, pp. 1028-104 May 2013.
2. FethiAbduljwad, Wang Ning, Xu De School of Computer and Information Technology Beijing Jiaotong University, 'SMXIR: Efficient way of Storing and Managing XML Documents Using RDBMSs Based on Paths', 2010 2nd International Conference on Computer Engineering and Technology,VOL.1 , pp. 143-147 2010.
3. GajananDnyanobaTemgire, Bharati Kale, Mitigating Duplicity in Hierarchical Data Using XML Mining, in International Journal of Advanced Research in Computer Science and Software Engineering, Vol No.4 Issue 11, pp 407-410, November 2014.
4. PavelShvaiko and JeromeEuzenat, Ontology matching: state of the art and future challenges, in IEEE Transactions On Knowledge And Data Engineering, pp. 158-176, Jan 2013.
5. ThandarLwin and ThiThiSoeNyunt University of Computer Studies, Yangon, Myanmar, 2010 Second International Conference on Computer Engineering and ApplicationsAn Efficient Duplicate Detection System for XML Documents, Computer Engineering and Applications, pp. 178-182 2010.
6. Ahmed K. Elmagarmid, Senior Member, IEEE, Panagiotis G. Ipeirotis, Member, IEEE Computer Society, and Vassilios S. Verykios, Member, IEEE Computer Society , Duplicate Record Detection: A Survey, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 19, NO. 1, pp. 1-16 JANUARY 2007.
7. L. Leitao and P. Calado, Duplicate Detection through Structure Optimization, Proc. 20th ACM Intl Conf. Information and Knowledge Management, pp. 443-452, 2011.
8. SairaGillani, Muhammad Naeem, Raja Habibullah, Amir Qayyum, Semantic Schema Matching Using DBpedia, in I.J. Intelligent Systems and Applications, Vol No. 04, pp. 72-80, 2013.