A New Approach to Broadcast XML Data in Wireless Environment Supporting Twig Pattern Queries

M.S.Namose
Computer Engineering
JSPM, NTC Narhe-41Pune,India
mahadevi.sggs@gmail.com

R.H.Kulkarni
Computer Engineering
JSPM, NTC Narhe-41Pune,India
rkpv2002@gmail.com

***Abstract*— In this paper we study and discuss a novel encoding technique of broadcasting XML data using twig pattern in wireless environment. That specific encoding method is characterized as Lineage Encoding. Lineage Encoding is a light-weight encoding scheme which is depends on a novel unit structure called G-Node for spilling XML information streaming in the wireless Environment. G-Node wipes out basic structural overheads of XML documents. The fundamental motivation behind this paper is to enchance energy and latency efficiency of broadcasting XML data in wireless environment. In lineage Encoding technique we have to use Structure Indexing and Attribute Summarization. Structure indexing captures the structural information of XML document. Attribute Summarization eliminates repetitive attribute names, in this way it reduces the size of a wireless XML stream.**

Index Terms* —*Twig pattern matching Structure Indexin, Attribute Summarizatio, Lineage Encodin, Twig Pattern

I. Introduction

Mobile computing is the core part of today's environment due to the advances of wireless technologies and increasing user demands. Wireless enables rapid and efficient information exchange with remote, mobile, or otherwise hard-to-reach people and equipment. Mobile Computing is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device .The main concept involves in mobile computing are mobile communication, mobile hardware, mobile software.

In the later versions of the Internet browsers support XML. In the near future, XML documents like HTML Web pages, become a part of our daily life. XML has attracted attentions from database community. All the major IT companies use XML into the software products. XML is used for describing how information is structured. This makes it much easier to move structured information from one program to another .XML performs a similar function when exchanging data which has a structure more complex than simply a string of text. XML has another important function, it doesn't specify how information on a particular topic should be structured, it directly provides a syntax for writing such specifications, called XML Schemas.

We need to consider energy conservation of mobile clients when disseminating data in the wireless mobile environment, because they use mobile devices with limited battery-power (i.e., energy-efficiency). The total query processing time must be reduced to provide fast response to the users (i.e., latency-

efficiency).In mobile computing reducing the battery consumption is a challenging one. There can be many reasons for power consumption but most noted one is, while using mobile internet that too especially while downloading data from the server. In present pull technology (client-server method), total content of the page is retrieved, because of this, content to be downloaded from a particular page is additional than required if you need to retrieve specific content then the location step is needed, which is complicated to remember. This causes more consumption of energy in mobile. Latency is also a major setback faced in the present method.

This problem can be overcome by push technology because when the client is waiting for the data it consumes less energy (doze mode) than when the data is downloaded from the server (Active mode) [14]. In Push technology the client will be subscribe to needed data provided by the server so whenever a new update available related to the data then it will be push to the client so in this case the transaction is initiated by the server thus client can receive the data without sending the request that consumes more energy. So it helps to minimize both power consumption and latency issues. Push technology is most preferably used for the data which has frequent update. E.g. push let, HTTP server push.

In this paper we study a streaming unit of a wireless XML data stream, called G-node. Also we study a light weight encoding technique, called Lineage Encoding, to show the parent-child relationships among XML elements in the G-nodes. We also study relevant functions and operators that exploit bit wise operations on the lineage codes. This is the first wireless XML streaming approach which supports twig pattern query processing in the wireless broadcast environment. Fig.1 is the architecture diagram for Lineage Encoding.

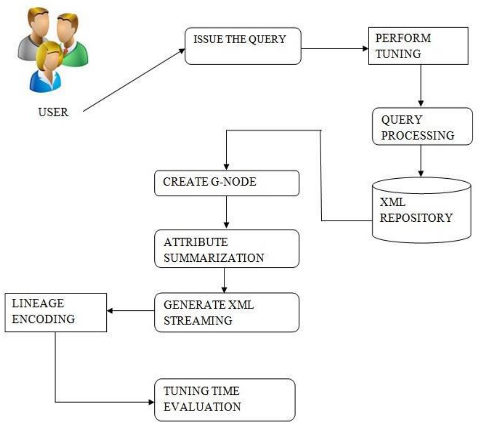

Fig. 1 Architecture Diagram For Lineage Encoding

II. BACKGROUND

A. Tree Pattern

There are two types of Tree Pattern [2].

*1) **Tree Patterns which support XML algebras:*** The XML tree algebra is a set of operators to manipulate data trees.

- TAX TP
- Generalized TP (GTP)
- Annotated TP
- Ordered Annotated TP

*2) **Tree Patterns Used in Optimization Processes :*** This Tree Pattern allow a broader range of queries

- Global Query Pattern Tree (G-QPT)
- Twig Pattern
- Logical Operator Nodes
- Node Degree and Output Node Specification
- Extended Formula
- Extended TP.

B. XML Query Language

XML query language is a markup language used for the documents contain structured information.XML is used to represents the structured and semi-structured documents[2]. Originally designed to meet the demanding of large-scale computerized broadcastin.XML is also used in the transaction of a wide variety of data. The expanding fame of XML is because of the confinements of the other two technologies are Hypertext Markup Language (HTML) and standard generalized markup languages (SGML)**.**

C. XML Data Model

XML documents have a hierarchical structure. These XML trees (twigs) are divided in two forms; first id ordered and another is unordered XML trees. The present approach considers an ordered and labeled XML tree. It can be represented as a rooted, ordered, and labeled tree.XML document must contain a root element. This element is the parent of all other elements. All elements in an XML document can contain sub elements, text and attributes. The tree represented by an XML document starts at the root element and branches to the lowest level of elements. The nodes of the XML tree represent elements and the edges represent parent-child relationships among XML elements[10].

D. Twig Pattern Query

A twig pattern query consists of two or more path expressions and it represents complex search condition.

E. Structure Indexing

As per author Jun Pyo Park, Chang-Sup Park, and Yon Dohn Chung "The structure indexing directly captures the structural information of XML documents and is used for XML query processing" Structure indexing creates summaries of the structure present in semi-structured data collections by grouping data items with similar structure, providing a mechanism to index such items. Since semi-structured data models are commonly represented by labeled graphs or trees (the XML data model being a prime example), structural indexes or summaries are naturally described as graphs where nodes represent sets of data items (called extents), and where edges represent structural

F. Attribute Summarization

As per author Jun Pyo Park, Chang-Sup Park, and Yon Dohn Chung "Attribute Summarization eliminates repetitive attribute names in a set of elements when generating a stream of G-nodes."

An attribute of an element consists of pairs of an attribute name and an attribute value. In addition, an element can contain a large number of attributes. There is a structural characteristic that nodes of same path use same attribute names, and thus the size of the XML stream can be effectively reduced by eliminating redundant attribute names[12].

G. Structured characteristics

Means that we can define exactly how the data is to be arranged, organized and expressed within the file. When we are given a file, we can validate that it conforms to a specific structure, prior to importing the data. As we know the structure of the file in advance, we know what it contains and how to process each item. Prior to XML, the only structure in a text file was positional – we knew the bit of text after the fourth comma should be a date of birth – and we had no way to validate whether it was a date of birth, or even a date, or whether it was in day/month/year or month/day/year order.

H. Drawbacks Of Previous Work

Conventional XML query processing methods cannot process XML twig pattern queries efficiently since they do not contain branching information or parent-child relationships. Previous work focus only on the problem of efficiency and scalability[5]. None of these methods focus on the energy-efficiency issue. Example: Multipredicate merge join (MPMGJN) algorithm[14]. Several approaches have been proposed for energy and latency efficient XML query processing in the wireless mobile environment. S-node generates an XML data stream based on the unit for XML broadcasting, called a S-node[15].

III. LINEAGE ENCODING

A. G-Node:

As per author Jun Pyo Park, Chang-Sup Park, and Yon Dohn Chung "Wireless XML stream is the combination of integrated G-Node". G-node integrates information of elements of the same path. Figure 2 shows the GroupNode structure constructed by structural summaries and attribute summarization. A GroupNode contains Group name, Location path, Child addresses, Attribute name list, Attribute value list, and text data. Group name is the tag name of integrated elements and Location path represents the label path of GroupNode. Child addresses are pointers to the child GroupNodes. Text address is pointer to its text list. Attribute name list is a sequence of pairs of attribute name and the address to its value list. Attribute value list contains a sequence of (attribute value, Dewey order) pairs. Finally text data is all contents of integrated elements[11].

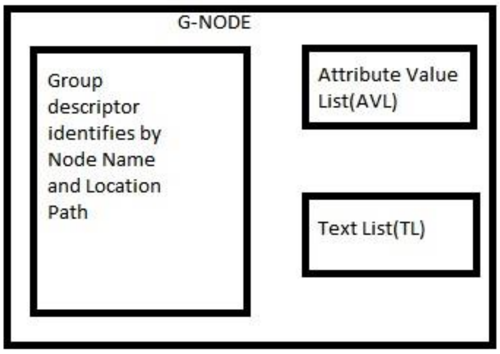

Fig. 2. G node Structure[11]

B. Lineage Encoding

As per author Jun Pyo Park, Chang-Sup Park, and Yon Dohn Chung "This is the first proposed wireless xml streaming approach. Lineage Encoding is a light-weight encoding scheme. This system consist of mainly two parts 1) vertical code denoted by Lineage Code(V) 2) horizontal code denoted by Lineage Code(H). This system support twig pattern query. Twig pattern is a one type of tree patterns used in optimization process. Twig Pattern query used to represent parent-child relationships among XML elements in two G-nodes."

In order to reduce the structural overhead of XML document, we extract tag name, attributes, and text by using SAX parser[10]. Stream generation can be considered a two-phase process. In the first phase, the stream generator constructs GroupNodes by extracting tag names and location paths. In the second phase, the stream generator performs attribute summarization and address calculation[12].

```
Algorithm XML_Data  Flow
Input is :XML document X
Output is:  Stream of  XML data  XS
begin
while (X != NULL)
if(p=PS)
Get GroupNode GN of path p
Add attributes, text, and dewey order into GN
else
Generates a new GroupNode GN
Insert tag name, location path, attributes, text, dewey order of
element into GN
Push GN into NodeStack NS
```

```
13. endwhile
14. while (NodeStack NS!=NULL)
15. Get top entry GN in NodeStack NS
16. Generate Attribute name list, Attribute value list and text list
17. Calculate Child addresses
18. endwhile
19. Flush all GN in NodeStack NS into the XML data stream XS
20. end
```

Fig. 3 XML stream generation algorithm[12]

Figure 3 is the the XML stream generation algorithm. At first, the stream generator makes GroupNodes by extracting tag name, attributes, and text of elements. If path of encountered element is not in path stack, the stream generator initialize a new GroupNode, otherwise it loads existing GroupNode of same path and merges attributes and text of element. In order to perform attribute summarization Dewey order is stored in GroupNode (Line 5-13). If the stream generator encounters the end of an XML document, the stream generator performs attribute summarization and calculates Child addresses, text addresses, and address in attirbute name lists (Line 15-19)[12].

IV. Twig Pattern Query Processing

As per author K. Preethi, S. Ganesh Kumar "Twig pattern query processing consists of three phases namely tree traversal phase, sub path traversal phase, and main path traversal phase. While processing a twig pattern query with predicates, we should select subset of elements satisfying the given predicates. Then, for the selected elements, we should find their parent elements or child elements. For example, to process the query in Fig. 4.

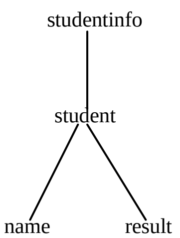

Fig. 4 Example twig pattern query

Q1://studentinfo/student[name/text()="predicate"]/result, we should find a subset of "name" elements satisfying the given predicate condition, select their parent "student" elements, and then identify "result" elements which are children of those "student" elements. A subset of the elements selected in a G-node can be represented by a bit string, called a selection bit string (SB) for the G-node where 1-value bits identify the selected elements. First, a function to obtain a selection bit string identifying a subset of elements in a particular child G-node is defined as follows:

Function1:

$$Vp=Shrink\&Mask(V,SBm) \quad (1)$$
$$SBn=Unpack(Vp,H)$$

A selection bit string SBn for child G-node N can be computed based on the Lineage Code of N (V, H) using Shrink&Mask and Unpack operators in order. Shrink&Mask (V,SBm), where V denotes LC (V) of child G-node and elements in m with one or more child elements in N are selected by SBm. Shrink&Mask operator computesVp by shrinking 0's in LC (V) and then it shrinks SBm by eliminating the bits in same position as those removed in V. Unpack(Vp, H), where Vp is the shrunken bit string computed by Shrink&Mask operator and H denotes LC (H) of child G-node. Unpack operator extends Vp based on H to obtain the result selection bit string for the G-node N. Thus a subset of elements in a particular child G-node is selected. Second, a function to identify the parent elements of a subset of elements selected in a G-node is defined as follows:

Function 2:

$$Vp=Pack(SBn,H) \quad (2)$$
$$SBm=Expand\&Mask(V,Vp)$$

To identify the parent elements of a subset of elements selected in a G-node, Pack and Expand&Mask operators are used. Pack(**SB_n,H**) operator computes **V_p** by shrinking the bit string **SB_n** based on H. V_p denotes the elements in the parent G-node of N which are parents of the elements in N selected by **SB_n** . Expand&Mask (V, V_p) operator expands V_p and masks V_p with it to obtain the result selection bit string SB_m for parent G-node of N. Thus selection bitstring to identify a subset of elements in a particular child G-node and to identify the parent elements of a subset of elements selected in a G-node is found. Finally, we define a function GetselBitst (J) to select elements in a Gnode contained in the query tree of a given twig pattern query, which satisfy all the branching paths and predicate conditions in the sub-tree. The selection bit string SBj for J can be computed by performing bitwise AND operations over all the selection bit strings SBm obtained from the child nodes of J where J is a G-node in the query tree T[4][10].

A . Sub path and Main path Traversal Phase

As per author K. Preethi, S. Ganesh Kumar **"**The main path denotes a path from the root node to a leaf node which represents the target element of the query and the sub path denotes branch paths excluding the main path in the query tree. The mobile client enters query and decryption key into the application. The query is then modeled into a query tree. In the tree traversal phase, the query tree is traversed in a depth-first manner; it selectively downloads components of the relevant G-nodes into the nodes in the query tree. Attribute values and texts involved in the given predicates are decrypted using the decryption key and downloaded into the relevant nodes. In the Subpaths traversal phase, the mobile client performs a postorder depth-first traversal starting from the highest branching node in the query tree using the GetSeleBitSt() function. In the sub path traversal phase each sub path is explored from the leaf node. Thus, the selection bit string for the branching node is calculated from all the sub paths in a bottom-up manner using Pack and Expand&Mask operators. Finally, the Main path traversal phase propagates the selection bit string on the branching node along the main path using Shrink&Mask and Unpack operators. Finally, the mobile client retrieves the required data which satisfies the given twig pattern query and with the help of encryption technique [4]". Figure 5 display the XML tree, this is the first step query processing. Figure 6 is the output screen of Lineage encoding technique, in which we show the number of nodes traverse.

Algo: Twig pattern Query Processing
Input: Wireless XML Stream XS, Twig pattern query TQ
Output: Result RE
begin
Initalize RE= ɸ
Selection bit string SB=1
Root G-Node of lineage code is(1,(1));
NextNode address denoted as DS;
//Step1: Tree Traversal phase
Construct a query Tree T for TQ
Repeat{
Collect a GD of G-Node which is denoted by NextNode;
IF(current node CN is the leaf node in T) then store AVL and TL the node in T;
Else
IF(CN contains predicate conditions P) then
Collect the relevant attribute values and text using AI and TI;
Store the relevant attribute values and text into node in T;
End if
Assing the address of the next node in CI to NextNode in T;
End if
} Until (all nodes in T are visited)
//Step2: Subpath traversal phase
N be the highest branching node in T;
SB_N=GetselBitst(N);
//Step3: Main Path Traversal Phase
MP be the main path in T starting from N;
P=N;
SB_p=SB_N;
Repeat{
Initialize C be the child node Of P in MP;
SB_c=SelChild(C,SB_p);
P=c;
$SB_{p=}$ $SB_{c;}$
} Until (C is the leaf node)
Select a set RE of elements in C using the selection bit string SB_c;
Return RE;

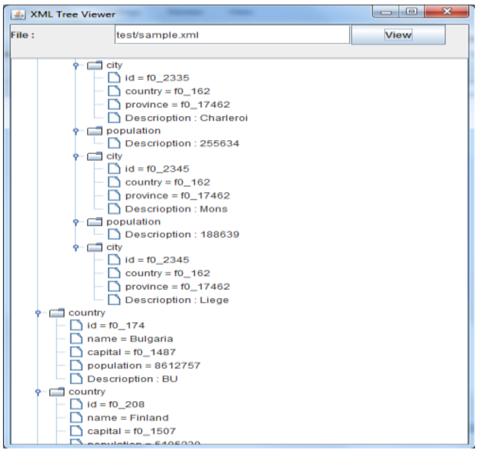

Fig. 5 XML Tree[10]

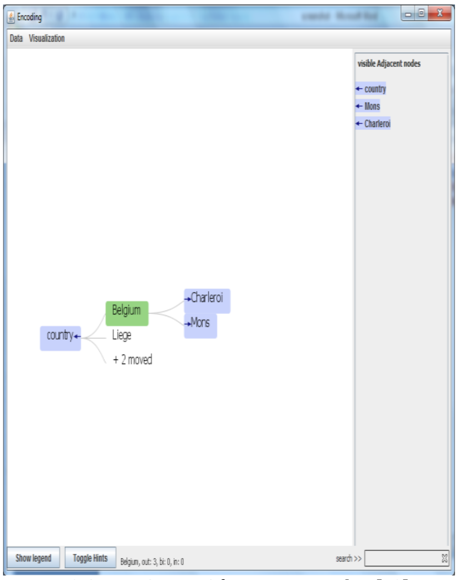

Fig. 6 Output Screen Of Lineage Encoding[10]

V. IMPLEMENTATION

At server side we use Java platform jdk environment and at the client side mobile device which supports of Android OS of any version and for broadcasting Wi-Fi router is included. Initially an xml document is created using XML Automation tool and it is automatically placed in an xml repository thereby relying on third parties is reduced. At the server side when the above Xml document is extracted from

the repository it performs structure indexing and Lineage encoding.
Fig 6 denotes the XML for Book publication. This XML file consists of author of the publication, name of the book and the year of the publication. Structure indexing performs and generates the indices by capturing the structural information of the XML document. Lineage encoding produced the bits of LCV and LCH of the respective XML document thereby supporting Twig pattern Queries. G node is the group node which is created by the presence of link in the connected elements. Then after all the process gets over this xml stream is broadcasted through the Wi-Fi router. And at the client side through the mobile device when a query
Q4://Author/name/year is passed by mobile device then it process through twig pattern query depicted in Fig.7 and its sample result is displayed in Fig.8

```
<Books>
<Author>Magpub
</Author>
<Name>Physics
</Name>
<Year>1951
</Year>
<Author>Tecpub
</Author>
<Name>Biology
</Name>
<Year>1985
</Year>
</Books>
```

Fig. 6 Sample XML for Book Publication[11]

Structure Indexing:
{route=1, number=4, area=7, avail=4}

Lineage Encoding:
{number-area=11-00, avail-avail=0100-1, route-area=1-7, avail-number=0100-1, avail-area=0100-1, area-avail=011-00, route-avail=1-4, route-number=1-4}

G node:
area -> [NodeName=area, LocationPath=/route/number/area, CI={area=tvt#avd, avail=yes#nil, number=21#22}, LC(V,H)=11-00, AVL=null, TL=null]number -> [NodeName=number, LocationPath=/route/number, CI={area=tvt#avd, avail=yes#nil, number=21#22}, LC(V,H)=1-4, AVL=null, TL=null]avail -> [NodeName=avail, LocationPath=/route/number/area/avail, CI={area=tvt#avd, avail=yes#nil, number=21#22}, LC(V,H)=011-00, AVL=null, TL=null]

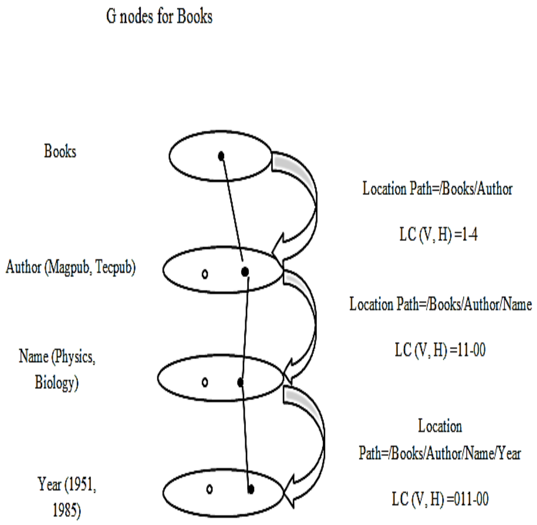

Fig. 7 Processing Query Q4 to get result

Publication Author: magpub

Book Name: Physics

Year of Publication: 1951

Fig. 8 Sample result after processing Q4[11]

VI. Conclusion

In this paper we study the push technology which helps to minimize the data download that reduce power consumption in mobile. Twig pattern queries containing complex conditions are popular and critical in XML query processing. In this paper, we study an efficient wireless XML streaming method supporting twig pattern queries. We study Lineage Encoding scheme and relevant operators to efficiently process twig pattern queries. This scheme explores , the benefits of the structure indexing and attribute summarization , so it reduces the size of the xml stream .The mobile client will retrieve the desired information by satisfying the given twig pattern and by performing the bit-wise operations on the Lineage Codes in the relevant G-nodes. This technique is effective and efficient in terms of the access time and tuning time. Because of their enormous records traditional XML inquiry processing methods are inefficient in the wireless mobile environment. In this way the global query processing time must also be reduced to provide fast response to the users.

In future, we plan to analyze the accompanying issues: First, depth-first traversal of components increases the access time for particular queries.

References

[1] W. Sun, Y. Qin, J. Wu, B. Zheng, Z. Zhang, P. Yu, P. Liu, J. Zhang " Air Indexing for On-Demand XML Data Broadcast" *IEEE Transactions* volume:25 , Issue: 6

[2] U. Ravindran, J. Sribhavani, "An efficient xml data streaming method in mobile wireless network " *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* volume 3, Issue 1, January 2014

[3] Y. Qin, H. Wang, J. Xiao "Effective scheduling algorithm for on-demand xml data broadcasts in wireless environments," *ADC* procedding of the twenty second Australasiam database conference-volume 115 page 95-102.

[4] K. Preethi, S. Ganesh Kumar "Secured wireless xml streaming supporting twig pattern queries," *International Journal of Computer Science and Information Technologies*, vol. 5 (2) 2014, 2112-211

[5] J. Pyo Park, C. Sup Park, Y. Dohn Chung "Lineage encoding: an efficient wireless xmlstreaming supporting twig pattern queries, **"**Knowledge and Data Engineering, *IEEE Transactions* on volume: 25 , Issue: 7 Publication Year: 2013 , Page(s): 1559 - 157.

[6] M. Hachicha , J. Darmont "A survey of xml tree patterns," *IEEE* transactions on knowledge and data engineering, volume. 25, no. 1, January 2013

[7] Y. Qin, H. Wang, J. Xiao **"** Effective scheduling algorithm for on-demand xml data broadcasts in wireless environments," *Australian Computer Society, Inc.22nd Australasian Database Conference(ADC 2011)*, Perth, Australia, January2011

[8] S. Vidya, R. Uma "Dynamic xml dissemination supporting twig pattern queries," *International Journal of Advanced Information Science and Technology (IJAIST)* ISSN: 2319:2682 Vol.17, No.17, September 2013.

[9] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: data management for asymmetric communication environments," Proc. *ACM SIGMOD* Int'l Conf. Management of Data Conf., pp. 199-210, Mar. 1995

[10] M. Namose, R. Kulkarni "Dissemination of xml data in wireless environment supporting twig pattern queries," *IJCSMC*, Vol. 3, Issue. 5, May 2014, pg.1061 – 1067

[11] J. Briskilal, D. Satish "An effective enactment of broadcasting xml in wireless mobile environment," *International Conference on Recent Trends in Information Technology*, 978-1-4799-4989-2/14

[12] J. Pyo Park, C. Sup Park, M. Kyung Sung, Y. Dohn Chung "Attribute summarization: a technique for wireless xml streaming," *ICIS* 2009, November 24-26, 2009 Seoul, Korea

[13] M. Francechet, "Xpathmark: an xpath benchmark for xmarkgenerated data," *Proc. Third Int'l Conf. Database and XML Technologies (XSYM)*, 2005.

[14] R. Goldman, and J. Widom, "DataGuides: Enable Query Formulation and Optimization in Semistructured Databases," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, pp. 436-445, 1997.

[15] T. Green, A. Gupta, G. Miklau, M. Onizuka, and D. Suciu " Processing xml streams with deterministic automata and stream index ," *ACM Trans*, Database Systems, vol. 29, no. 4, pp. 752-788, 2004.

[16] XMark - An XML Benchmark Project, http://www.xmlbenchmark. org, 2003.

[17] XML Data Repository, http://www.cs.washington.edu/research/xmldata sets, 2012.

[18] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman,"On supporting containment queries in relational databasemanagement systems," Proc. ACM SIGMOD Int'l Conf. Management of Data Conf. pp. 425-436, 2001.