

Querying XML documents with XPath/XQuery in presence of XLink hyperlinks

LULE AHMEDI and MENTOR ARIFAJ
University of Prishtina, Computer Engineering
Kodra e Diellit p.n.
10000 Prishtinë, Kosova
{lule.ahmedi | mentor.arifaj}@fiek.uni-pr.edu

Abstract: Nowadays XML documents are presented into the Web more than ever. Using XLink hyperlinks, a document can refer different portions of information in different documents into the Web. With XLink hyperlinks the XML documents in the Web can be investigated from querying point of view not just while browsing these documents. There are few implementations regarding XLink recommendation and just two implementations regarding querying XML documents using XPath/XQuery expressions in presence of XLink hyperlinks. On the other side XPath/XQuery has many implementations and wide support from IT leaders in the world. In this paper it is shown another way of using XPath/XQuery to query XML documents in presence of hyperlinks. The paper shows a model and its implementation. To materialize the new model it is used Saxon-B processor, with which was created an extension named BOTA. Prototype BOTA can be used in different web applications and easily can be integrated into Web Services in order to select something more narrow than the whole XML document. The prototype BOTA especially can be used in applications with RSS and Atom.

Key-Words: - XML, Querying XML, XLink hyperlinks, XML Applications, LDAP.

1. Introduction

Today XML documents into the Web are present more than ever before. These XML documents are not just standalone documents but using XLink hyperlinks these documents refer each other. In this architecture where files refer each other querying is a necessary in order to find right information and remains in an evolving phase. Despite the fact that XLink is a W3C recommendation since 2001 there are few implementations [17] for it. Based on the investigations there are two implementations that querying XML documents in presence of XLink hyperlinks using XPath/XQuery expressions. On the other side there are wide range of implementations regarding XPath/XQuery and these implementations are in different platforms. For XPath/XQuery could be said that there are support from IT leaders on the world as many of them included XPath/XQuery in their database management systems [18].

1.1 Structure of the paper

Section two presents the current hyperlinks technologies. On it are mentioned HTML, XPointer and XLink as main hyperlinks technologies today on the Web. Af-

terward are mentioned issues left open after XLink recommendation specially querying XML documents in presence of XLink hyperlinks. In order to make possible to query XML documents using XPath/XQuery expression in presence of XLink hyperlinks are proposed several possible access.

Section three describes the proposed model. First it was presented that it could be used expressions of XPath `id()` function via XQuery expressions. But due to drawbacks of the approach it was followed another kind of approach. The model makes aware XPath/XQuery processors for presence of XLink hyperlinks.

Section four presents the implementation of the model using Saxon-B processor creating new prototype named BOTA.

And at the end in section five, we discuss about related work and conclusion of our implementation that it is not a duplication of the existing implementations but a complementary implementation for query interlinking XML documents.

1.2 Previous papers

There is a previous paper related to the same model. The paper [22] presents the main part of the model, i.e.

awaring XPath/XQuery processors for presence of XLink hyperslinks.

2. Hyperlink technologies and querying issue

XLink or XML Linking Language is a recommendation from World Wide Web Consortium since 2001. But there are several technologies that are related to links into the Web. In this section there are described several hyperlinking or related technologies and further more are discussion about querying XML documents in the presence of XLink hyperslinks.

2.1 HTML

HTML or HyperText Markup Language is a language basen on SGML with elements into HTML document starting with a tag "<" and are ending with tag ">". HTML offers a fixed vocabulary of elements in order to describe documents in Web[6] whereas XML offers a replacement of these fixed elements of HTML with a set of elements which are specified in a specific application[16].

HTML hiperlinks are realized in that way that an element of a document refers into whole another document. After creating relations between documents HTML hiperlinks can be used to refer resources such as video, figure, audio file [6]. These references are one-direction hyperslinks [3].

Using an anchor into a document, a part of the HTML document could be refert from other HTM document but still you can have the whole content of the document.

HTML hyperslinks today are widely used in the Web and also are simple to use. On the other side HTML hyperslinks have some drawbacks among others [7]:

- Hyperslinks are incorporated into HTML documents and cannot be saved outside the documents
- HTML hyperslinks do not offers semantic between relations of the documents
- HTML hyperslinks create links between two documents

2.2 XPointer

With XPointer could be selected more detail content than XML node level. XPointer could be considert as an extended form of XPath. XPointer it is used with other recommendations like XLink for realizing hyperlings between resources [15].

2.3 XLink

XML Linking Language (XLink) is WorldWideWeb Consortium (W3C) recommendation since 27 June 2001 and offers a possibility to create one way hyperslinks and multi-directional hyperslinks.

XML elements that are hyperslinks could be into the same XML document and also could be in diferent documents into the Web.

For XLink hyperslinks it could be said:

- Hyperslinks are designed for use with XML
- Hyperliks could be multi-directional
- Every XML element could be a hyperslink not just a predefined element such as "a" element of HTML

XLink offers two types of hyperslinks:

- XLink simple hyperslinks
- XLink extended hyperslinks

These two types of hyperslinks support the needs for the Web from creating simple hyperslinks like HTML and to complicated and sophisticated hyperslinks structure between documents into the Web.

From the database point of view the Web with hyperslinks could be seen as a distributed database with hyperslinks connecting different porcions of the database objects and it could be used to realize queries into it. For realizing querying actual technologies are in development or evolving phase [6].

There exist needs where distributed XML documents into the Web should be presented as an integrated document and a big XML document should be separated into the different parts into the Web. In these cases the hyperslinks are playing a significant role connecting XML documents and querying these documents is very necessary in order to select only the wanted information.

To distinguish the type of the elements there exist a type attribute and with "simple" value the XML element forms a simple XLink hyperslink whereas the type attribute with "extended" value within XML element forms an extended XLink hyperslnk. An extended hyperslink plays a role as a container for XML sub-elements within it in order to create the whole extended hyperslink. These sub-elements could be with values of the attribute type with a value "locator", "resource", "arc" or "title".

XML data into the Web could be seen from two points of view. One is from the Web browsers and the other point is from the databases. On this paper the XML data are seen from the database point of view and how to use these data into different applications.

2.4 Issues after XLink

XLink defines standard way of creating hyperlinks into XML documents. But, even XLink is recommendation since 2001 there are some issues left open and there aren't any working draft or recommendation from W3C regarding these issues. At the work of [5] are mentioned some issues left open after XLink but here could be mentioned these two:

- Presenting hyperlinks while reading with Web browsers
- Realize querying XML documents in presence of XLink hyperlinks

From the point of view of this paper important is issue of querying XML documents in presence of XLink hyperlinks. It is known that XPath/XQuery realize querying withing a XML document and there are no any response from W3C how to deal with this issue with XML documents in presence of XLink hyperlinks.

XLink has few implementations these can be seen from official web site [17] of W3C for XLink implementations. Often these implementations are more oriented while browsing rather than dealing with issue of querying XML documents. This fact could be describe that XLink can define a powerful platform for creating hyperlinks into XML documents and are defined concepts and mechanisms that are not used enough. But in the Web data are presented more and more due to change of the Web concept from presenting data into the Web to a distributed powerful database with data widely distributed into the Web. So, after that there is a need how to manipulate and how to select presented data into the Web. Due to wide use of HTML hyperlinks and its simplicity it could be said that there was a kind of inertia of use new hyperlinks defined by XLink.

XPath/XQuery has many implementations and these implementations are in different platforms. There exist implementations with open source and commercial uses. Also there are implementations into Database Management System such as SQL Server 2005, Oracle and DB2 this can be concluded directly via official web site of the recommendation from W3C [18].

Based on our investigation there are two systems that deals with querying XML documents using XPath or (and) XQuery in presence of XLink hyperlinks. These systems are:

- Extension of LDAP system and
- Extension of eXist system

Lightweight Directory Access Protocol (LDAP) presents a system that uses other data model rather than XML data model so based on this LDAP system could

be considered as non native XML system. Using LDAP it was done the extension [1, 2] in order to query XML documents with XLink hyperlinks using XPath.

Extension of eXist system was done based on logical model proposed by Wolfgang May and his colleagues [9]. eXist is a native XML database written in Java and on it are included many recommendations such as XPath/XQuery, XSL, XML Schema, XInclude [6].

2.5 Existing approach

In order to make possible to execute query express with XPath/XQuery expression and for XML documents containing XLink hyperlinks we have several ways to follow:

- Using XInclude to merge distributed XML documents in a single XML document
- Using distributed queries to query distributed XML documents
- Using logical model described by May and his colleagues [9]

2.5.1 Using XInclude

XInclude is a XML based technology with aims to merge different XML documents in a single document.

From the perspective of querying XML documents in presence of XLink hyperlinks it is possible to use XInclude in that way to bring to a single server content of distributed XML documents in a single document and to execute query expressed with XPath/XQuery expression without the need to extend XPath/XQuery and for XLink hyperlinks.

XInclude distinguish from XLink in that XLink has not specified any processing model but using attributes offers mechanisms for detecting hyperlinks, whereas XInclude defines specific processing model for join informations. So, for realizing the merging of XML documents using XInclude it is necessary to have a XInclude processor. Also, it would be necessary another processor for execution of query expressed with XPath/XQuery expressions. If it is used this access it is necessary to have two processors one for XInclude and another for XPath/XQuery. Also there is a question how will function two processors together. Based on XInclude [10] recommendation, XInclude has no relation to DTD and XML Schema and this makes difficult validation of new XML document constructed after merging of distributed XML documents.

At the end, although existence of XInclude technology and merging possibilities of the XML documents distributed on different servers and connected via XLink

hyperlinks, in the database perspective this technology has several drawbacks. As a drawback could be absence of a data model [6], and as another drawback could be that XInclude cannot express reciprocal relations between recourses [4] and so it is not comfortable in the database environment.

The other drawback in the point of view of our aim is the complicated architecture of two different processors if we use XInclude for the aim of extending XPath/XQuery and for XLinks, too.

2.5.2 Using distributed queries to query distributed XML documents

The term of distributed query means a query that will return a result from distributed XML documents. Realizing of distributed queries could be done in different way such as:

- Bringing all data from different XML documents in a place where is processor and then execution of the query
- Distributing the query into the locations of XML documents and then return the result

It is known that functions `fn:doc()` and `fn:put()` that are defined from XPath/XQuery recommendation brings data from distant server where is located XML document and makes possible execution of XPath/XQuery expression using a processor. From paper [19] there exist an implementation which use a number of XQuery processors that are located into different servers and together participating into execution of queries. This is done using Extended Remote Procedure Call (XRPC) that is based on SOAP communication protocol that in fact it is based on XML and with this brings to integration of XQuery with Web Services. Using this method will lead to a complicated architecture with distributed XPath/XQuery processors and these processors could not be located into whole Web servers participating into Internet. So, from the point of view of extending XPath/XQuery for XLink hyperlinks we see this as a drawback.

2.5.3 Using Logical model proposed by May and his colleagues:

The logical model proposed from Wolfgang May and his colleagues [9] is an extension of XLink from the database perspective of XML documents in presence of XLink hyperlinks. The model adds new attributes other than XLink recommendation's attributes. With new semantic it is made possible to query XML documents containing XLink hyperlinks with XPath/XQuery ex-

pressions. The model was materialized extending eXist a native XML database system.

3. Model: Explicitly following hyperlinks

As XPath/XQuery is dealing with addressing elements within a XML document and also XPath/XQuery has a widely support from the IT community, our primary goal of this paper is to demonstrate a way how to extend XPath/XQuery to query XML documents connected via XLink hyperlinks.

As distributed XML documents could be connected using attribute `xlink:href` defined by XLink recommendation, this attribute generally it is expressed using expression 1.

Expression 1: $xlink:href = uri-reference$

Further more expression 1 could be expressed into two parts the url part and the fragment part as in expression 2.

Expression 2: $uri-reference = url\#fragment$

The url is Unique Recourse Locator that could be expressed via expression 3.

Expression 3: $url = scheme://host:port/path$

From the expression 2, the fragment part in general could be XPointer or other expression. In this paper the intention is to demonstrate the extension of XPath/XQuery for XLink hyperlinks, it was found the first condition.

Condition 1: $fragment = xpathexpression$

This condition we will treat only cases where the fragment is XPath or in general and XQuery expressions. This condition it is based on papers [1, 2] that it is used during the extension of LDAP for query XML documents in presence of XLink hyperlinks.

The extension of XPath/XQuery for XLink hyperlinks could be done trying to extend `id()` function of XPath with XQuery expressions.

Expression 4: $fragment = path1/id(parameter)/path2$
XPath `id()` function could be expressed using XQuery expressions like in expression 5.

Expression 5: $path1/id(parameter)/path2 = (for \$priid in (path1)$

$for \$posid in$
 $(/path2/parent::node())$
 $where$
 $\$priid/parameteri=\$posid/attribute::node()$
 $return (\$posid)$

where $\$priid$ and $\$posid$ are XQuery variables.

Although the expression 5 could be taken into account for extending XPath/XQuery and for XLinks but also it

contains some drawbacks. Among otherse these drawbacks are:

- Expression 5 already exists
- Partial extention, just for id() function

So, if it would continue the extention basen on expression 5, it would be considert more like a simulation of something that already exist rather than a contribution that ofering new approach.

3.1 XPath/XQuery processor awareness about XLink hyperlinks

XML documents could be present with data model with tree of nodes. The nodes could be child or attribute of an XML document. With presence of XLink hyperlinks in XML documents the tree data model could be expanded in model of oriented graph with circles. It is yet possible for XPath/XQuery expressions to navigate between nodes of the graph [6].

Definition 1: XML data model in presence of XLink hyperlinks is not a tree but it is a graph with interconnected nodes of children/attributes and additional type of arcs that allow to refere to local or distant part of nodes in the graph.

Connections between XML documents distributed into the Web are realized using the xlink:href attribute.

Condition 2: For execution of query expressed with XPathXQuery expression for XML documents in presence of XLink hyperlinks nodes that contains href attribute must be presented into that query with the @xlink:href attribute

Conditions 1, 2 and definition 1, present the base of new model of extention of XPath/XQuery for XLink hyperlinks.

Condition 2 is based on logical model of presenting XML documents with definition 1 and shows that for connecting with documents of distant resources, it is used xlink:href attribute during formulation of query and with this the processor will follow the attribute axis for realizing the query.

Condition 2 could be illustrated with expression 6.

Expression 6:

//xpath1/xpath2/.../xpathk[@xlink:href]/.../xpathn

Part xpathk of the query in expression 6 tells the processor to follow the attribute axis to proceed with processing the query.

To understand more the new model especially the condition 2 it is presented the algorithm which deals with extention of XPath/XQuery processors for XLink hyperlinks based on conditions 1 and 2 and also on definition 1.

Algorithm (ProcessingXLink(S))

Let S be a query expression

Let B be an indicator of the presence of (@xlink:href)

// Initialize B to False

B = False

If (S contains (@xlink:href))

Let S1 = S (part before @xlink:href with added "/string(@xlink:href)")

Let S2 = S (part after @xlink:href)

Let B = True

Else

// everything remains as in Saxon-B

End If-Else

If (B == True)

Let A be the array of results from processing of S1, which could have 0,

1 or more members with values of @xlink:href attribute,

*/**

Array A content could be like:

A[0] = "http://.../file0.xml#xpathexpression[0]"

A[1] = "http://.../file1.xml#xpathexpression[1]"

...

A[n-1] = "http://.../file(n-1).xml#xpathexpression[n-1]"

n is the total number of members of A)

or in general A[i] = uri[i]#xpathexpression[i]

**/*

For (i = 0, i < n, i++)

A[i] = A[i] & S2

A[i] = Separate(A[i])

ProcessingXLink (A[i])

End For

End If

Figure 1: ProcessingXLink procedure part of the extention of XPath/XQuery for XLink hyperlinks

Also in figure 2 it is presented the simple algorithm of deviding the url part from fragment part of the value in xlink:href attribute.

Algorithm Separate(C)

Let C be a string of the form C = url#xpathexpression

Let U = url // first part of argument C

Let X = xpathexpression // second part of argument

return X

Figure 2: Separate procedure for deviding url part from fragment part of value in xlink:href attribute

Algorithmes in figures 1 and 2 shows the logical way of how to be precesses the XPath/XQuery expression in presence of XLink hyperlinks.

Example 1: Mondial database [11] countires.xml file adopted for the intention of query XML document in presence of XLink hyperlinks with XPath/XQuery expression of querie. The url part of the value of xlink:href attributes was substituted in general form like url = http://localhost/Mondial-Distributed/.../filename.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE countries SYSTEM "countries.dtd">
<countries
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:dbxlink="http://dbis.informatik.uni-
    goettingen.de/linxis">
  <country car_code="AL" area="28750">
    <name>Albania</name>
    <population>3249136</population>
    <population_growth>1.34</population_growth>
    <infant_mortality>49.2</infant_mortality>
    <gdp_total>4100</gdp_total>
    <gdp_agri>55</gdp_agri>
    <inflation>16</inflation>
    <government>emerging democracy</government>
    <ethnicgroups
      percentage="3">Greeks</ethnicgroups>
    <ethnicgroups
      percentage="95">Albanian</ethnicgroups>
    <religions percentage="70">Muslim</religions>
    <religions percentage="10">Roman
      Catholic</religions>
    <religions percentage="20">Albanian
      Orthodox</religions>
    <capital xlink:type="simple"
      dbxlink:transparent="make-attribute insert-nodes"
      xlink:href="http://localhost/Mondial-
        Distributed/Cities/cities-
        AL.xml#/cities/city[name='Tirane']"/>
    <cities xlink:type="simple"
      dbxlink:transparent="drop-element insert-nodes"
      xlink:href="http://localhost/Mondial-
        Distributed/Cities/cities-AL.xml#/cities/city"/>
```

```
<encompassed xlink:type="simple"
  dbxlink:transparent="group-in-element insert-nodes"
  xlink:href="http://localhost/Mondial-
    Distributed/continents.xml#/continents/continent[name
      ='Europe']"
  percentage="100"/>
  <neighbor xlink:type="simple"
    dbxlink:transparent="duplicate-element insert-bodies"
    xlink:href="http://localhost/Mondial-
      Distributed/countries.xml#/countries/country[@car_c
        ode='GR']"
    borderlength="282"/>
  <neighbor xlink:type="simple"
    dbxlink:transparent="duplicate-element insert-bodies"
    xlink:href="http://localhost/Mondial-
      Distributed/countries.xml#/countries/country[@car_c
        ode='MK']"
    borderlength="151"/>
  <neighbor xlink:type="simple"
    dbxlink:transparent="duplicate-element insert-bodies"
    xlink:href="http://localhost/Mondial-
      Distributed/countries.xml#/countries/country[@car_c
        ode='YU']"
    borderlength="287"/>
</country>
<country car_code="GR" area="131940">
  <name>Greece</name>
  ...
</countries>
```

The new proposed model and the algorithm will be illustrated with the case of query Mondial database an XML database [11] with different XML documents connected with XLink hyperlinks.

Expression 7: Query Mondial database to find all Albanian cities. The query based on contions1 , 2 and definition 1 is:

$S = \text{"http://localhost/Mondial-Distributed/countries.xml\#/country[@car_code='AL']/cities[@xlink:href]/name"}$
Based on algorithm, the expression 7 will be devided into two parts. Part before @xlink:href attribute adding and function string(@xlink:href).
Expression 8: $S1 = \text{"//country[@car_code='AL']/cities/string(@xlink:href)"}$

The function string(@xlink:href) it is added with the aim that the query S1 to return the values of the attribute in the attribute notes that query takes into account.

Remain part of the query S is given with the expression 9.

Expression 9: $S2 = "/name"$

In presence of attribute `xlink:href` the B indicator from the algorithm in figure 1 will tell the processor for presence or not of the hyperlinks and in that way the processor will continue in attribute axis or not. The result form the query in expression 8 will be an array with `xlink:href` attributes' values. These values will be saved into $A[n]$ array. The array will contain zero, one or more members depending from the values of the attributes and from the query itself.

From the expression 8 the array will contain one member with value in expression 10.

Expression 10: $A[0] = "http://localhost/Mondial-Distributed/Cities/cities-AL.xml\#\#/cities/city"$

Furthermore the execution of the query from expression 7 will continue with taking into account and remaining part of the main query. This can be expressed with expression 11 and 12

Expression 11: $A[0] = A[0] \& S2$

Expression 12: $A[0] = http://localhost/Mondial-Distributed/Cities/cities-AL.xml\#\#/cities/city/name$

Now the processing will continue with the algorithm Separate shown in figure 2 which is called from the algorithm ProcessingXLink from figure 1.

With the algorithm Separate will have the result in expression 13.

Expression 13: $A[0] = "/cities/city/name"$

Query expressed with expression 13 will be executed in the `cities-AL.xml` and as there are no attribute `@xlink:href` there will be the query in a XML document just like XPath/XQuery without XLink hyperlinks. The result of the expression 13 is shown in figure 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Tirane</name>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Shkoder</name>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Durres</name>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Vlore</name>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Elbasan</name>
<name
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:dbxlink="http://dbis.informatik.uni-
goettingen.de/linxis">Korce</name>
```

Figure 3: Result from execution of query from expression 7

The recursive form of the algorithm expressed in figure 1 makes possible execution of the expression of the query in more general form like in expression 14.

Expression 14:

```
//xpath1/xpath2/.../xpathk[@xlink:href]/.../xpathl[@xli
nk:href]/
.../xpathm[@xlink:href]/.../xpathn
```

From the mean that an XML document can refer to another XML document and these referred XML documents can refer to other XML documents, the `xpathk` and `xpathl` could be in different XML documents that tell explicitly processors to follow attribute axes instead child node.

3.2 Extreme cases of querying and the possibility of eliminating them from the proposed logical model

XML data model in presence of XLink hyperlinks can be expressed using graph data model that can contain cycles. The cycles mean that two nodes can refer to each other. These cycles are not desired while querying XML documents. There exist some extreme cases of querying that are so called "pathological cases". These pathological cases are elaborated in more details in [9] and we summarize:

- Link bombs
- Oscillators
- Infinite horizont grow

It is expected to be treated these pathological cases from the new proposed model of extending XPath/XQuery for XLink hyperlinks. A detail treated of these cases it is not yet done and it can be seen as a future work.

4. Implementation of the model

In order to materialize the new data model that extends XPath/XQuery for XLink hyperlinks, it is used well-known processor Saxon-B[13]. Saxon-B is an open source processor written from company Saxonica which offers and a commercial version named Saxon-SA, too. This processor is realized in Java and using IKVM .NET [8] tool it is possible to use Saxon in Microsoft .NET and Mono. Saxon-B is one of pioneer implementation of XSLT 2.0, XPath 2.0 and XQuery 1.0[23] and now it could be finding with version Saxon 9.1. Difference between Saxon-B and Saxon-SA is that Saxon-SA is a schema awareness processor and more detail features of these versions could be found in official web site of the Saxonica [14].

Saxon-B it is known as in-memory processor that means that for query XML document first the document it is loaded in memory and then executes the query. Like Saxon-B are Xalon and Galax processors, too.

Another feature of Saxon-B is that it is in group of processors that do not require being in server, it means non-server based processor [6] and it could be used to query XML files in a stand alone computer or in the Web. As a stand alone processor Saxon-B can be used easily in different applications.

4.1 Extention of Saxon-B

Realizing extention of Saxon-B to support XPath/XQuery expressions and for XLink hyperlinks it is used the source code of saxon-recources9-0-0-1[13] version. Extention is done based on condition 1, 2 and definition 1. The code is added into Saxon-B based on algorithm in figures 1 and 2.

The extention of Saxon-B it is named BOTA (from Albanian means "World") and this prototype was realized using NetBeans 6.0.1[12].

Where it is used from application prototype BOTA looks like in figure 4.

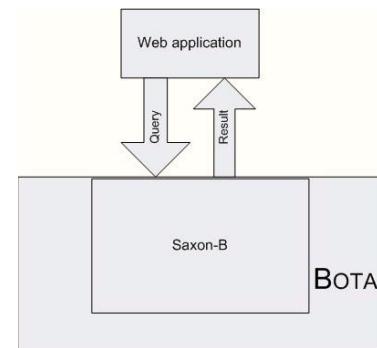


Figure 4: Prototype BOTA

5. Further work and conclusion

In the previous chapters are given the description of the model and implementation for query XML documents with XPath/XQuery expressions in presence of XLink hyperlinks.

5.1 Related work

It is said that new model it is based on conditions 1, 2 and definition 1. Condition 1 makes limitation of use of XPath query instead XPointer and other expressions that could be used in fragment part of the uri expression during creation of hyperlinks. Condition 2 makes the awareness of the XPath/XQuery processors for the presence of XLink hyperlinks. The extention of the XPath/XQuery processors is given with algorithm in figure 1.

Based on investigations there are already two implementations that extend XPath/XQuery to process and XLink hyperlinks in XML documents.

On implementation is extension of LDAP system [1, 2] to execute XPath queries in XML documents in presence of XLink hyperlinks. For this extention it is said that XPath queries are mapped to the data model that uses LDAP. The data model used for LDAP is non XML data model. This extention is important that makes possible access and query of XML files into different locations like in servers of an organization or servers into Internet. The difference between model used for extention of LDAP and new model is clear. The new model is an XML based data model whereas LDAP as it is said uses other data model than XML.

The model proposed from Wolfgang May and his colleagues [9] it is based on XML data model. In they model they extend XLink with new semantics with new attributes that makes possible for processors to follow XLink hyperlinks during processing the XML documents. They materialize their model extending eXist an XML database system. As both extentions are native XML,

the differences can be seen in the way how they follow hyperlinks and where to use their implementations. New model proposed in section 3, follows hyperlinks explicitly whereas the other model follows hyperlinks implicitly via the added attributes. Also the differences can be seen on the implementations. Extended eXist can be used on XML database systems whereas BOTA can be used in Web applications.

5.2 Futur works

The prototype BOTA has just came out and to achieve the final goal i.e. using prototype BOTA in the real Web applications it will need to consider some future works.

These works can be categories in

- technical works
- future works

5.2.1 Technical works

In this category of works are works that are related almost directly to the prototype BOTA.

As one of the works could be seen the feasibility of the prototype BOTA in the real Web. In the illustrated examples in the session 3 it could be seen that the preliminary tests for the prototype BOTA is done using XML documents distributed into a local network, whereas testing the feasibility of the BOTA into Web remains as a nearest technical work.

Furthermore the tests are done using XLink simple hyperlinks and as a planned test could be done using XLink extended hyperlinks, it is clear that the proposed model could be changed or could be more extended while taking into account and extended XLink hyperlinks.

Also as a technical work are considered treating the pathological cases. As the oriented graph data model can contain cycles between nodes it is desirable that the prototype BOTA to detect these cycles and to inform the applications that the formulated query enters into a cycle between two or more nodes.

5.2.2 Future works

As the prototype BOTA it is built using Saxon-B processor, the processor which is easily adoptable in different Web applications and as the initial tests show that the prototype is functional, BOTA can be used to be integrated into Web applications and Web Services [21]. In Web, BOTA will make possible to select something narrow that the content of the XML document. It

is possible to use prototype BOTA for RSS [20], to select hyperlinks with a criteria and not the whole content. Although it is easy to say, the prototype BOTA can be used on future search engines in the Web.

5.3 Conclusion

The contribution of the extension described in this paper will be summarized as follows:

- There is a new model for extending XPath/XQuery for XLink hyperlinks and is an alternative model of existing models
- An implementation with extension of Saxon-B processor, the extended processor is named BOTA
- Except Saxon-B there are many implementations of XPath/XQuery that could follow the new model in order to support and XLink hyperlinks during query XML documents with XPath/XQuery expressions.

Even the XLink recommendation is since 2001, from the perspective of database and applications the recommendation is in evolving phase.

XML documents can be everywhere into the Web and with XLink hyperlinks can create relations between these documents in order to refer meaning content without the need to create redundant data where the same data can be found into the Web. These hyperlinks in case of querying XML documents can tell processors the way where should be find another XML document to continue the process of querying.

Except two already mentioned models the paper presents one new approach of extending XPath/XQuery for XLink hyperlinks. The model is so formulated that the query should contain the attribute @xlink:href with the nodes that contain the attributes in order to aware the processor for the presence of XLink hyperlinks.

Based on preliminary tests the prototype and the model it is shown that is functional and could be considered for the technical and future works in order to achieve the maturity and it use into the Web.

References:

- [1] L. Ahmedi, *Global Access to Interlinked XML Data Using LDAP and Ontologies*, PhD dissertation from Albert-Ludwigs, Freiburg, 2001
- [2] L. Ahmedi, *Making XPath Reach for the WebWide Links*, SAC'05 Santa Fe, New Mexico, USA, ACM 1581139640/05/0003, 2005
- [3] F. Bry and M. Eckert, *Processing Link Structures and Linkbases in the Web's Open World Linking*,

- HT'05, Salzburg, Austria. ACM 1595931686/05/0009, 2005
- [4] E. Behrends, Evaluation of Queries on Linked Distributed XML Data, *PhD dissertation from Mathematical-Natural Science of George-August, University of Gottingen*, 2006
- [5] M. Eckert, Processing Hypertext Links after XLink, Project Work within the frames of the Advanced Practical Work, *Institute of Informatics, University of Ludwig Maximilians, Munchen*, 2004
- [6] O. Fritzen. Modeling and Querying of Distributed XML Data in Presence of 3rd Party Links, *PhD dissertation from Mathematical-Natural Science of George-August, University of Gottingen*, 2007
- [7] K. Gwiazda and P. Kazienko, Information Systems Architecture and Technology ISAT 2001. *Proceedings of the 23rd International Scientific School. Grzech A., Wilimowska Z. (eds), Wrclow, pp. 132-139*, 2001
- [8] IKVM.NET, *Java into Microsoft .NET and Mono*, <http://www.ikvm.net/>.
- [9] W. May, E. Behrends and O. Fritzen. Integrating and Querying Distributed XML Data via XLink, *Information Systems, to appear*, 2008
- [10] J. Marsh and D. Orchard, *XML Inclusions (XInclude) Version 1.0, W3C Recommendation 20 December 2004*.
- [11] Mondial database, <http://www.dbis.informatik.uni-goettingen.de/Mondial/#XML>
- [12] NetBeans 6.0.1, www.netbeans.org
- [13] Saxon-B source code web site: <http://saxon.sourceforge.net/>
- [14] Saxonica: <http://www.saxonica.com/>
- [15] J. Simpson, XPath and XPointer, *O'Reilly, ISBN: 0-596-0029-2*, 2002
- [16] E. Wilde, D. Lowe. XPath, XLink, XPointer, and XML: A Practical Guide to Web Hyperlinking and Transclusion, *Addison Wesley, ISBN: 0-201-703440*, 2002
- [17] W3C XLink implementation's web site: <http://www.w3.org/XML/Linking>
- [18] W3C XPath/XQuery implementation's web site: <http://www.w3.org/XML/Query/>
- [19] Y. Zhang and P. Boncz, XRPC: Interoperable and Efficient Distributed XQuery, *VLDB'07, Vienna, Austria, ACM*, 2007
- [20] G. Vossen, S. Hagemann, *Unleashing Web 2.0: From Concepts to Creativity*, Morgan Kaufmann, 2007
- [21] Alonso, G., F. Casati, H. Kuno, V. Machiraju: *Web Services. Concepts, Architectures and Applications*. Springer Verlag, Berlin, 2004
- [22] L. Ahmedi and M. Arifaj, Processing XPath/XQuery to be Aware of XLink Hyperlinks, *Proceedings of the 2nd EUROPEAN COMPUTING CONFERENCE (ECC'08)*, ISBN: 978-960-474-002-4, pp. 217-221, Malta, September 2008
- [23] M. Kay. XPath(tm) 2.0 Programmer's Reference, *eISBN: 0-764-57756-5, Wiley Publishing, Inc.*, 2004