XPointer

Project work Jindong Gu

Outline

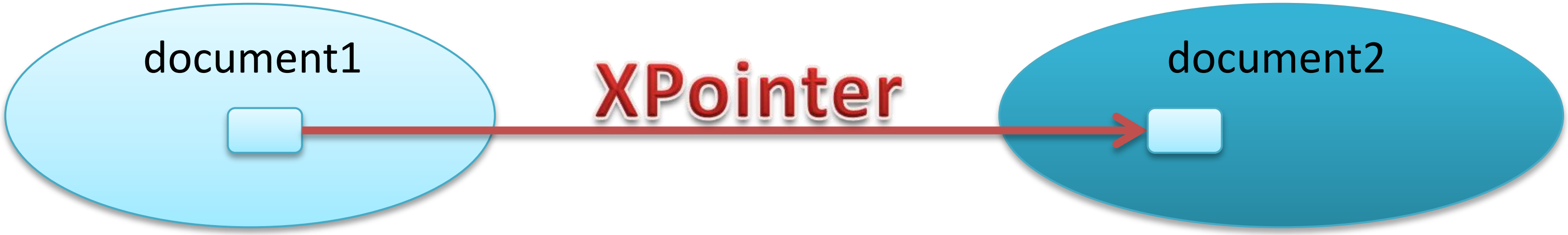
1. Introduction: Origin and Overview
2. XPointer Framework
3. Xpointer xlmns() scheme
4. Xpointer element() scheme
5. Xpointer xpointer() scheme
6. Applications in related technologies

1.The origin of XPointer

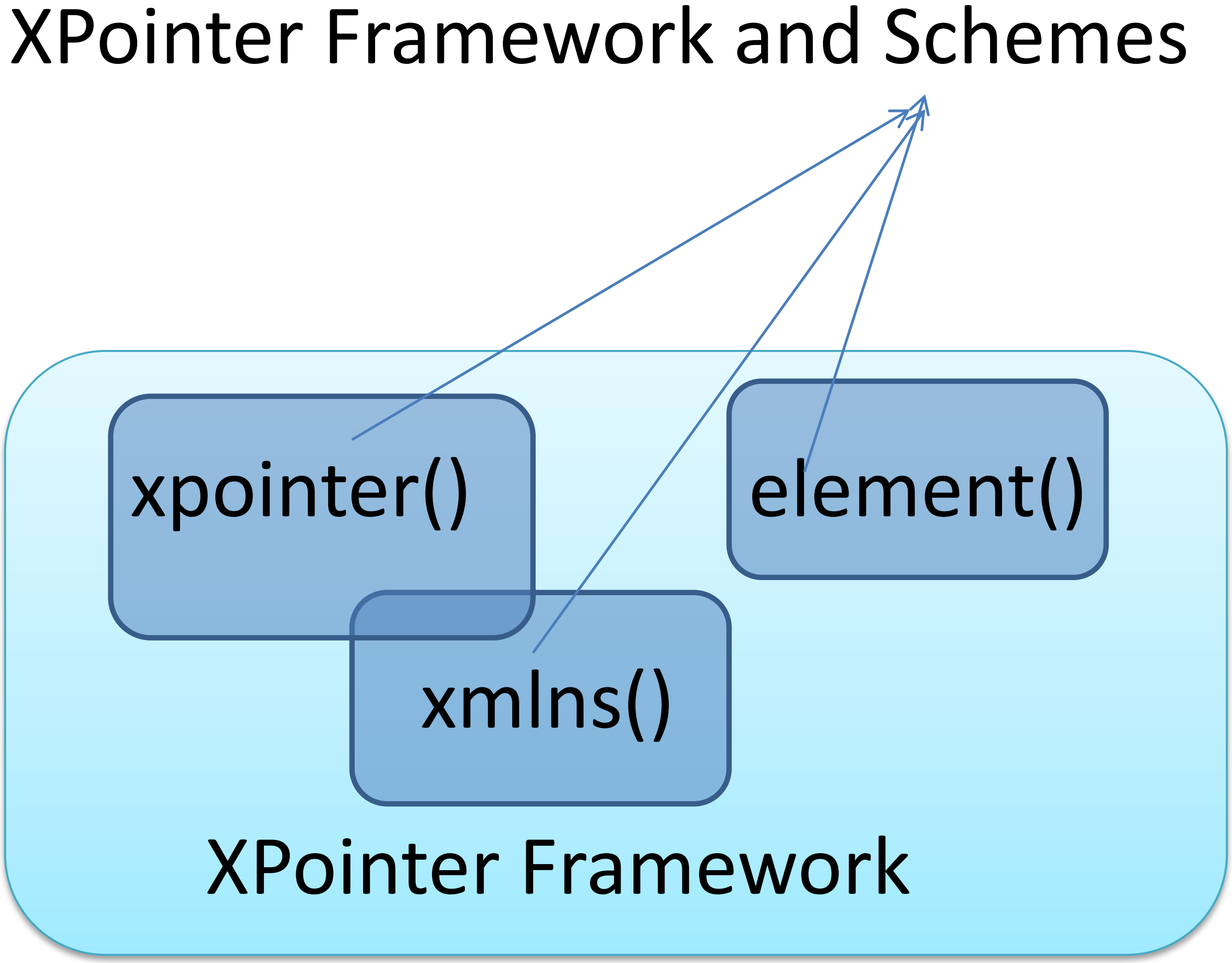
With the widespread deployment of XML documents, the identification of parts of documents is of great significance.

XPointer provides ways to identify any node or selection in any XML document by describing its structure and context.

XPointer uses XPath expressions to navigate in the XML document.



1.The overview of XPointer



2.XPointer Framework

- XPointer Framework is an extensible system for XML addressing that underlies additional XPointer schemes[1].

- The framework is intended to be used as a basis for fragment identifiers for any resource whose Internet media type is XML-based, e.g. ***text/xml, application/xml****.*

- The framework defines two types of pointers:

  - shorthand pointers

  - scheme-based pointers

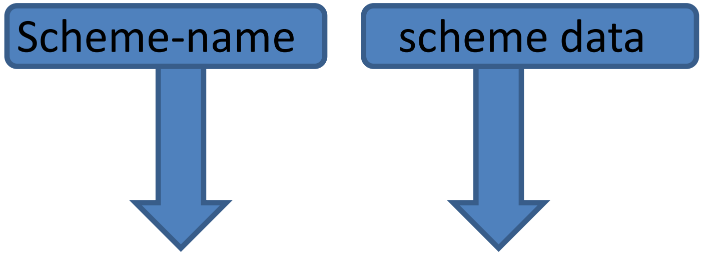
2.XPointer Framework

Shorthand Pointer:

book.xml#chapter_one

The xpointer identifies an element in” *books.xml*”, whose id is ”chapter_one”.

Scheme-Based Pointer:



Scheme-name

scheme data

book.xml#xpointer(id(”chapter_one”))element(chapter1)

This xpointer identifies an element in books.xml , whose ID is “chapter_one "or, if this cannot be found, it identifies elements whose name is "chapter1“.

3.XPointer xmlns() Scheme

The XPointer xmlns() scheme is intended to be used with the XPointer Framework to allow correct interpretation of namespace prefixes in pointers, for instance

- namespace-qualified scheme names
- namespace-qualified element and attribute names appearing within scheme data[2].

3.XPointer xmlns() Scheme

A pointer part of the xmlns() scheme declares a namespace prefix to be associated with an XML namespace name.

```
<?xml version="1.0" encoding="UTF-8"?>
<app  xmlns="http://app.com/a_info">
    <version xmlns="http://app.com/v_info">version 1.0</ version >
</app >
```

The following two pointer parts allow the xpointer() part to refer to the app element (as a:app) and the name element (as v:version).

```
#xmlns(a= http://app.com/a_info)xmlns(v= http://app.com/v_info) xpointer(/a:
app /v:version)
```

4.XPointer element() Scheme

The XPointer element() scheme is intended to be used with the XPointer Framework to allow basic addressing of XML elements[3].

Two patterns of applying element() scheme:

①without element name, e.g. *#element(/3/1)*

②with element name, e.g. *#element(name/2/1)*

The element name is not qualified because the element() scheme does not support the namespace binding context.

4.XPointer element() Scheme

XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<app  xmlns="http://app.com/a_info">
      <version xmlns="http://app.com/v_info"><num>version 1.0<num></ version >
      <version xmlns="http://app.com/v_info"><num>version 2.0<num></ version >
</app >
```

XPointer element() Scheme:

#element(app/2/1)

The above pointer identifies an element by first locating the element identified by the name “app", then locating that element's second child element, then finally identifying that element's first child element, namely, the “num” element.

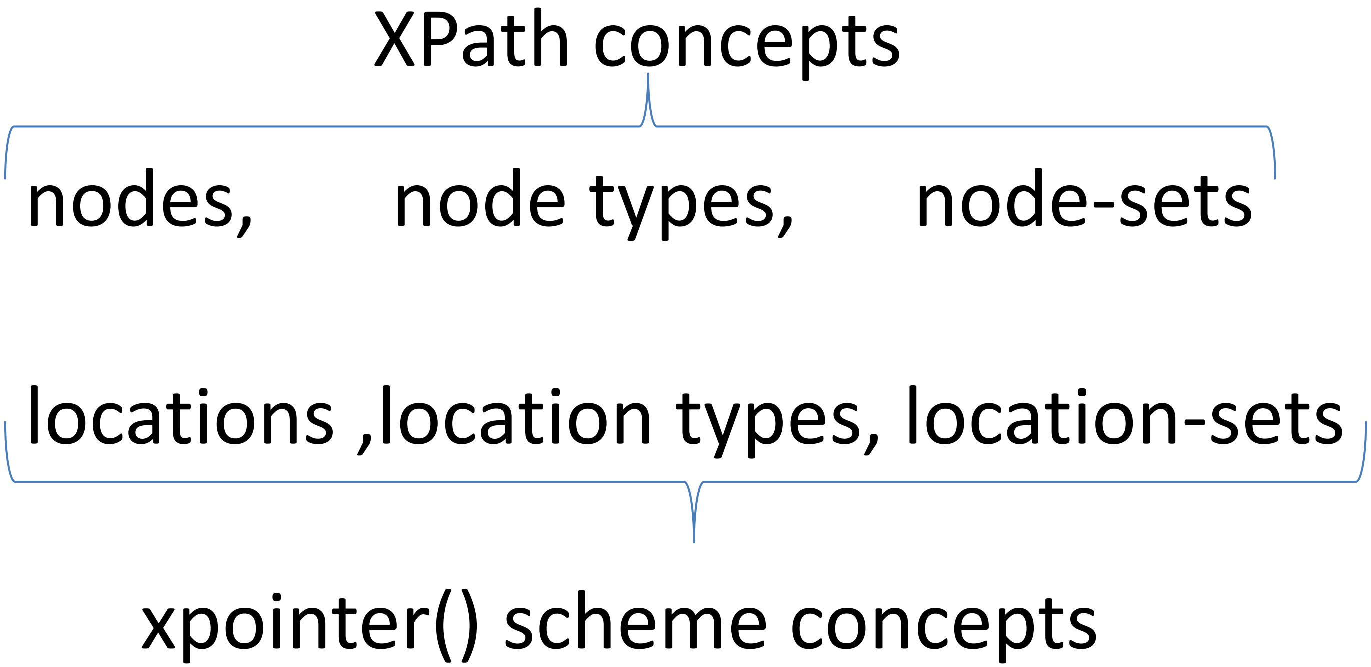
5.XPointer xpointer() Scheme

- The XPointer xpointer() scheme is intended to be used with the XPointer Framework to provide a high level functionality for addressing portions of XML documents[4].

- It uses XPath expressions to navigate in the XML document and adds functionality based on XPath to address strings, points, and ranges.

5.XPointer xpointer() Scheme

Compared to XPath, XPointer xpointer() scheme adds:



XPath concepts

nodes, node types, node-sets

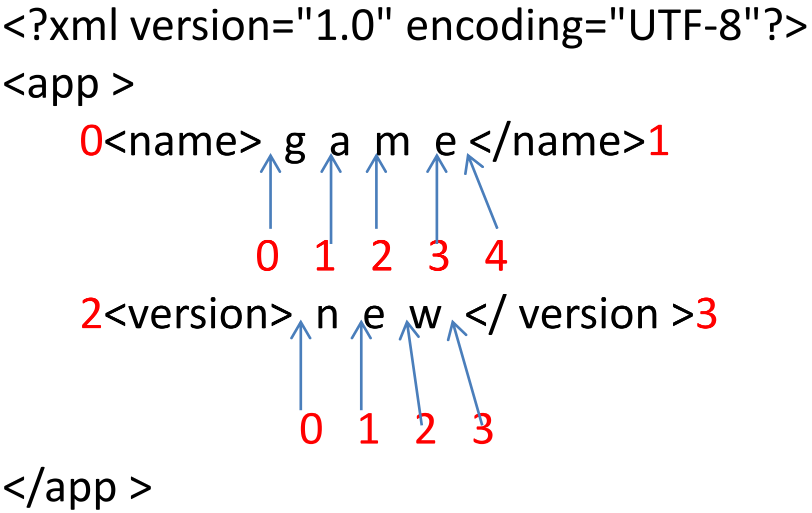
locations ,location types, location-sets

xpointer() scheme concepts

Two new location types: point and range

5.XPointer xpointer() Scheme

XML document:



```
<?xml version="1.0" encoding="UTF-8"?>
<app >
    0<name> g  a  m  e </name>1
           0  1  2  3  4
    2<version> n  e  w </ version >3
               0  1  2  3
</app >
```

Point is defined in terms of two data items, container node and non-negative index. The above described points are node points und character points.

Range is defined by two points, a start point and an end point. A range represents all of the XML structure and content between the start point and end point.

5.XPointer xpointer() Scheme

Besides, xpointer() scheme has added some functions related to point, range :

Point related: start-point(), end-point(),

Range related: range-inside(), range-to(), covering-range(), string-range().

A function *here()* is applicable when the xpointer() scheme expression being evaluated is found within an XML document (→).

A function *origin()* is applicable when a xpointer() scheme expression is interpreted in the course of a hypertext link traversal (→).

5.XPointer xpointer() Scheme

Use of function here():

XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<app xmlns:xlink="http://www.w3.org/1999/xlink">
    <newVersion>version 1.0</newVersion >
    <version xlink:type="simple“
               xlink:href="#xpointer(here()/preceding::newVersion[1]”>
    new version</version>
</app >
```

The pointer identifies the attribute that contains the function here(), and then selects all nodes that appear before the current attribute node in the document. Finally the first “newVersion”-element will be selected.

5.XPointer xpointer() Scheme

Use of function origin():

XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<app xmlns:xlink="http://www.w3.org/1999/xlink">
   <version xlink:type=“locator“  xlink:label=“v"  xlink:href="http://app.com/v1.xml “>
            version 1.0</version>
   <version xlink:type=“locator“  xlink:label=“v"  xlink:href="http://app.com/v2.xml ">
            version 2.0</ version >
   <version xlink:type=“locator“  xlink:label=“v"  xlink:href="http://app.com/v3.xml “>
            version 1.0</version>
   <redirect xlink:type="locator" xlink:label=“redirect“
                          xlink:href="#xpointer(origin()/following-sibling::version[1])" />
   <next xlink:type="arc" xlink:from=“v“ xlink:to=“redirect" />
</app >
```

The relative pointer with origin() sets out from the begin of the link, and then selects the next “version”-element with the rest part of it.

5.XPointer xpointer() Scheme

Identify a point:

```
<?xml  version=”1.0”>
<Wuppertal>
<faculty1  id=“DMT”> print and media technology </faculty >
<faculty2  id=“IT”>internet technology</faculty >
<faculty3  id=“BWI”>enterprise economy law </faculty >
< /Wuppertal>
```

The expression #xpointer(start-point(//faculty1)) identifies the point between “>” of <faculty1 id=“DMT”> and “p” of print and media technology.

The container node of this point is the “faculty1”-element and the index of it is zero.

end-point() funktioniert analog.

5.XPointer xpointer() Scheme

Identify a range:

```
<?xml  version=”1.0”>
<Wuppertal>
<faculty1  id=“DMT”>print and media technology </faculty >
<faculty2  id=“IT”>internet technology</faculty >
<faculty3  id=“BWI”>enterprise economy law </faculty >
< /Wuppertal>
```

The expression #xpointer(id(“IT")/range-to(id(“BWI")))
returns a range from the start point of the element with id=“IT” to the end point of the element with id=“BWI”;
That range marked by red color in XML document will be identified.

5.XPointer xpointer() Scheme

Identify a String value:

```
<?xml  version=”1.0”>
<book> Andersen's fairy tales, which have been translated into more than 125
languages, have become culturally embedded in the West's collective consciousness,
readily accessible to children, but presenting lessons of virtue and resilience in the
face of adversity for mature readers as well.
</book>
```

The expression *string-range(//book,“125 languages",5,9)[1]* returns a string value, whose start point immediately precedes the letter “l" (5th from the start of the string) in the first string " 125 languages "appearing in a book element, and its length is 9.

XPointer

The applications of XPointer :

Many types of XML-processing applications need to address into the internal structures of XML resources using XPoiter and URI references, for example,

1.XML Linking Language [XLink],
2.XML Inclusions [XInclude],
3.Resource Description Framework [RDF],
4.SOAP 1.2 [SOAP12],
5.Extensible Stylesheet Language Transformation[XSLT2.0].

6.1.Application of XPointer with XLink

XML Linking Language allows attributes to be inserted into XML documents in order to create and describe links between resources[5].

Based on XLink, XPointer can identify not only standard fragments of URIs with the aim to create and describe links between any kinds of resources.

6.1.Application of XPointer with XLink

Target XML document to be linked:

```
<?xml version="1.0" encoding="UTF-8"?>

<auto  relation=“”sale”  time=“2014”>
   <Mercedes>
       <quantity_of_sale>min 84000 max 86000</quantity_of_sale>
   </Mercedes>
    <BMW>
       <quantity_of_sale>min 74000 max 80000</quantity_of_sale>
    </BMW>
    <Porsche>
       <quantity_of_sale>min 44500 max 50000</quantity_of_sale>
     <Porsche>
</auto>
```

The URI of this XML document is “http://www.autoinfo.com/sale1.xml”

6.1.Application of XPointer with XLink

XML Document:

```
<?xml version="1.0" encoding="UTF-8"?>
< BMW  xmlns:xlink="http://www.w3.org/1999/xlink">
   <compare xlink:type="simple“
     xlink:href="http://www.autoinfo.com/sale1.xml"></compare>
   <sale>
            <sale time=”2013 ” >79000</sale>
            <sale time=”2014 ” xlink:type="simple“
xlink:href=“http://www.autoinfo.com/sale1.xml#xpointer(string(//BMW,text(),5,5)[1])”/>
            <sale time=”2015”>no result</sale>
   </sale>
</BMW>
```

The first link identifies the preceding entire XML document without XPointer .

The second link identifies a part of resource(a certain string value) form preceding XML document with XPointer.

6.2.Application of XPointer in XInclude

XML Include is devised to construct a single XML document from multiple documents.

XInclude supports inclusion of XML documents and non-XML text documents and allows authors to control the recovery process[6].

XInclude supports not only inclusion of entire documents, but also of parts of document by providing XPointer pointers , which point to part(s) of a XML document.

6.2.Application of XPointer in XInclude

Target XML document to be included:

```
<?xml version="1.0" encoding="UTF-8"?>

<auto  relation=“”sale”  time=“2014”>
   <Mercedes name=“Mercedes”>
       <quantity_of_sale>min 84000 max 86000</quantity_of_sale>
   </Mercedes>
    <BMW name=“BMW”>
       <quantity_of_sale>min 74000 max 80000</quantity_of_sale>
    </BMW>
    <Porsche name=“Porsche”>
       <quantity_of_sale>min 44500 max 50000</quantity_of_sale>
    <Porsche>
</auto>
```

The URI of this XML document is “http://www.autoinfo.com/sale2.xml”

6.2.Application of XPointer in XInclude

XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<BMW xmlns:xi="http://www.w3.org/2003/XInclude >
        <compare>
                <xi:include  href="http://www.autoinfo.com/sale2.xml“ />
        </compare>
        <saleinfo>
              <xi:include href="http://www.autoinfo.com/sale2.xml "
                      xpointer="xpointer(//BMW[@name='BMW’])"/>
        </saleinfo>
</BMW>
```

The instruction includes entire preceding XML document.

The instruction includes a part of document (BMW element) by providing XPointer pointer.

6.3.Application of XPointer in RDF

The **Resource Description Framework** (**RDF**) , originally designed as a metadata data model, has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources[7].

Resource identification in RDF:
The subject, predict, object of an RDF statement can be uniform resource identifiers(URI). The URI can point at a partial resource of a xml with XPointer pointers.

6.3.Application of XPointer in RDF

XML document *(“http://www.person.com/info.xml”)* :

```
<?xml version="1.0" encoding="utf-8"?>
<persons>
   <person id="person1“><name>Bob</name></person>
   <person id="person2"> <name>Jill</name></person> </persons>
```

RDF document:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:dc="http://purl.org/DC/">
 <rdf:Description  rdf:about="http://rama.cpe.fr/index.html">
    <dc:Creator rdf:resource=" http://www.person.com/info.xml #xpointer(id(person1))"/>
     <dc:Title> Application of XPointer in RDF </dc:Title>
  </rdf:Description>
</rdf:RDF>
```

The instruction indicates certain a resource as an object of relationship that predicate “Creator” represents.

6.4.Application of XPointer in SOAP 1.2

SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages[8].

SOAP is based on XML.
When the web server respond to a soap request, it can reply the requested information by providing a URI with a XPointer pointer.

6.4.Application of XPointer in SOAP 1.2

BERGISCHE UNIVERSITÄT WUPPERTAL

Web resource:

This is a XML document in the web.

```
<?xml version="1.0" encoding="UTF-8"?>
<auto  relation=“”sale”  time=“2014”>
      <BMW id=“BMW”>
           <quantity_of_sale>min 74000 max 80000</quantity_of_sale>
      </BMW>
      <Mercedes id=“Mercedes”>
           <quantity_of_sale>min 84000 max 86000</quantity_of_sale>
     </Mercedes>
       <Porsche id=“Porsche”>
           <quantity_of_sale>min 44500 max 50000</quantity_of_sale>
       <Porsche>
</auto>
```

The URI of this document is “http://www.autoinfo.com/sale3.xml”

6.4.Application of XPointer in SOAP 1.2

A SOAP request:

```
POST /InStock  HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
         <soap:Body xmlns:car="http://auto.example.org/">
                    <car:GetSaleInfo>
                    <car:name>BMW</car:name>
                  </car:GetSaleInfo >
         </soap:Body>
</soap:Envelope>
```

6.4.Application of XPointer in SOAP 1.2

A SOAP response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset="utf-8“
Content-Length: nnnn
<?xml version='1.0' ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Body>
      <car:SaleInfoResponse  xmlns:car="http://auto.example.org/">
      <car:sale auto:source= " http://www.autoinfo.com/sale3.xml #xpointer(id=“BMW”)“/>
       </car:SaleInfoResponse >
</soap:Body></soap:Envelope>
```

The response message includes the element car:sale by providing a attribute value of URI with XPointer pointer, from which the (ultimate) receiver could retrieve the resource as a part of the message.

6.5.Application of XPointer for XSLT 2.0

XSLT (Extensible Stylesheet Language Transformations) is a language for transforming XML documents into other XML documents or other formats such as HTML for web pages, plain text or into XSL Formatting Objects[9].

During transformation one or more elements ought to be identified by XPointer pointers to match templates.

6.5.Application of XPointer for XSLT 2.0

Target XML document to be transformed:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl"   href=“auto.xsl“ ?>
<autos>
  <carname>
   <car>
       <name>Mercedes</name>
   </car>
    <car>
       <name>BMW</name>
    </car>
    <car>
       <name>Porsche</name>
     </car>
  </carname>
</autos>
```

6.5.Application of XPointer for XSLT 2.0

XSLT Document [9](“auto.xsl”):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0“  xmlns:xsl=“http://www.w3.org/1999/XSL/Transform”
xmlns:xptr=“http://www.rotorz.net/xml/2009/Xpointer”>
<xsl:template match=“autos">
      <html>
          <head> <title>Car name</title> </head>
           <body><ul> <xsl:apply-templates select=“carname"/> </ul> </body>
     </html>
</xsl:template>
<xsl:template match=“carname”>
     <xsl:for-each select="xptr:lookup-node(./car)”>
          <li><xsl:value-of select="name"/> </li>
      </xsl:for-each>
</xsl:template>
```

Output as following:

```
Mercedes
BMW
Porsche
```

The value of select attribute can be XPointer pointers , which identify elements of corresponding XML document.

XPointer

Quelle:

[1].Paul Grosso, Arbortext , editor. XPointer FrameworkW3C Recommendation 25 March 2003.

[2].Steven J , editor. DeRose XPointer xmlns() SchemeW3C Recommendation 25 March 2003.

[3].Paul Grosso, Arbortext , editor. XPointer element() Scheme, W3C Recommendation 25 March. 2003

[4].Steven DeRose , editor XPointer xpointer() Scheme,W3C Working Draft 19 December 2002.

[5].Steve DeRose, Eve Maler, and David Orchard, editors. *XML Linking Language (XLink)*. World Wide Web Consortium, 2001.

[6]Ionathan Marsh and David Orchard, editors. *XML Inclusions (XInclude) Version 1.0*. *Work in progress.* World Wide Web Consortium, 2001.

[7].Dave Beckett, editor. *RDF/XML Syntax Specification*. World Wide Web Consortium, 2001.

[8].Nilo Mitra et al., editors. *SOAP Version 1.2* Parts 0, 1, and 2. World Wide Web Consortium, 2001. *Work in progress.*

[9] XPointer Framework for XSLT 2.0 http://rotorz.com/xml/2009/XPointer/