

Свързване и обединение на XML съдържание – XLink, Xpointer и XInclude



Същност
Типове
Елементи
Обхождания
Примери

Въведение

- Доброто структуриране на XML документа предполага по-високо ниво на грануларност на информацията.
- Това води до необходимост от свързване на едни секции на документа с други.
- Секции от съдържанието трябва да могат да реферират към съдържание във външни документи по начин, по-ефективен от хипервръзките в HTML.
- Ограничения при използването на традиционните HTML хипервръзки – напр. задължителната посока на връзката от съдържащия я документ към външен ресурс или фрагмент в документа.
- Това води до развитие на стандарти за свързване и обединение на XML съдържание.

Три стандарта за свързване и обединение на XML документи

- **XLink** – език за дефиниране и описание на различни видове хипервръзки между Уеб ресурси, в частност XML документи. Възприет е като стандарт през 2001 година.
- **XPointer** – език за дефиниране на хиперлинкове в XML документ, които обаче реферират различни фрагменти от XML документа. XPointer дефинира фрагментни идентификатори за URI референции, сочещи към възли или части от тях в XML ресурси. Стандарт от 2002 година.
- **XInclude** - общ механизъм за сливане на XML документи, посредством задаването на маркери (тагове) в главния документ за автоматично включване на други документи или части от тях. Стандарт от 2006 г. насам.

Xlink, XPointer и XInclude

- Xlink, XPointer и XInclude използват:
 - **XPath** за адресация и навигация в XML документи
 - стандарта **XML Base** (<http://www.w3.org/TR/xmlbase/>), който дефинира атрибута **xml:base** за задаване на база за относителни URI връзки към външни за документа ресурси, подобно на елемента HTML BASE.
- XPointer е предназначен за използване с XML Link Language, или съкратено XLink. Използва относително ограничено в световен мащаб.
- Както XML Schema и XPath, така и XLink и XPointer са **спомогателни езици (XML accessories)**, които разширяват свойствата на езика XML. Специално за XLink и XPointer, разширението е в посока дефиниране на различни видове хипервръзки между документи и в самия документ.
- XML Inclusions (XInclude) е по-известен и въвежда общ механизъм за сливане на XML документи за използване от софтуерни приложения. XInclude дефинира език за конвертиране на входни XML данни в друго крайно представяне. Ето защо XInclude, заедно с CSS и XSL, се отнася към семейството на **XML преобразувателите (XML transducers).**

Какво е XLink?

- XLink е предназначен за описание и създаване на връзки между Уеб ресурси и е специално разширение на XML за хипермедия.
- XLink преодолява недостатъците на хипервръзките между HTML документи.
- Съкращение от: **XML Linking Language (XLink)**
- Версия: 1.0 (*by W3C XML Linking Working Group*)
- Спецификация: <http://www.w3.org/TR/xlink/>
- Namespace:

xmlns:xlink="<http://www.w3.org/1999/xlink>"

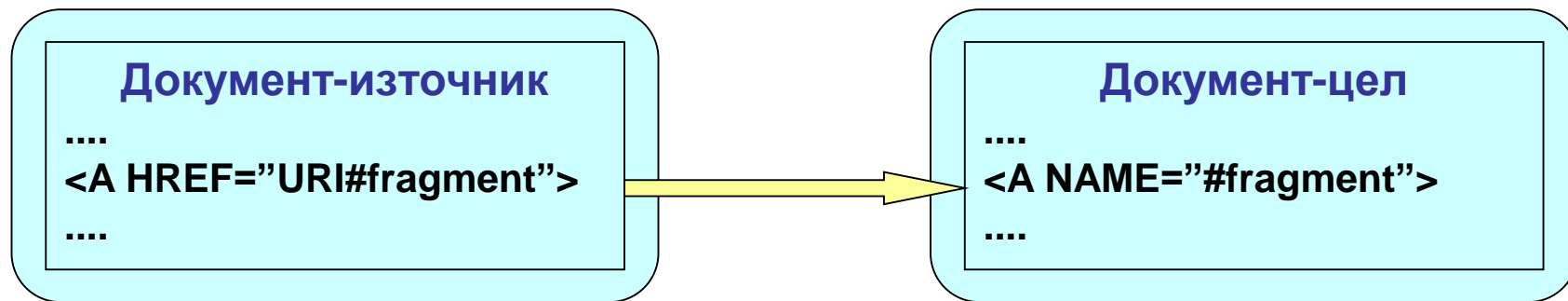
HTML хипервръзка 1/2

В HTML, връзката се изгражда на базата на елемента за котва с маркер **<A>** и се разпознават по това име, което не може да послужи за универсален механизъм за свързване в XLink. Това е така, защото котвата може да се използва по два начина:

- За да създадем връзка в един документ (източник) към друг (цел), чрез използване на атрибута HREF;
- За създаване на показалец (*bookmark*) в документа-цел с помощта на име атрибута NAME (*ако целта не е целият документ-цел, а фрагмент от него*).

HTML хипервръзка 2/2

- Така реално елементът HTML котва не представлява самата връзка, макар че е наричан с това име.
- За една хипервръзка в общия случай трябва да се създадат две котви, и то на две различни места, като дефиницията на връзката трябва да е непременно в документа-източник.



Абстрактна семантика на връзката

- XLink задава на езиково ниво абстрактна семантика на връзката, като източникът и целта се задават *относително*, а не *абсолютно*.
- Както хипервръзките в HTML, така и в XLink може да се дефинират прости и еднопосочни връзки.
- Освен това: ->

XLink може да се използва за:

1. дефиниране на връзки между повече от два ресурса – целта може да е повече от една;
2. дефиниране на връзки не само в документа източник, но и в документа цел;
3. задаване както на вътрешни връзки (вградени в документите), така и на дефинирани на външно място, отделно от свързаните ресурси – използва се за случаите, когато не е възможно да се дефинират връзки в ресурси като графика, видео и аудио;
4. свързване на метаданни с XLink връзки.

XLink концепции 1/2

- Според спецификацията (IETF RFC 2396), **XLink връзката** се посочва изрично от XLink елемент на свързване (*linking element*), който утвърждава съществуването ѝ.
- Една връзка (линк) е елемент и асоциира явно набор от ресурси или части от ресурси, за които се казва, че участват в тази връзка.
- **Ресурс** е всяка адресируема единица информация или услуга - файл или фрагмент от него, изображения, програми и резултати от заявка. Биват:
- **Локални (*local*) ресурси** – вътрешни за документа, представляващи елемент на свързване на дадена връзка или неговия родител;
- **Отдалечени (*remote*)** – представляват както външни за документа ресурси, така и вътрешни ресурси (напр. елементи), достъпни по URI.

XLink концепции 2/2

- Идентификаторът на отдалечен източник, участващ във връзката, се нарича **локатор** (*locator*).
- Използването на връзка за минаване по нея за каквато и да е цел се нарича преминаване или **траверс** (*traversal*). Източникът, от който е започнал траверсът, се нарича **начален ресурс**. Дестинацията на преминаването през връзката е **краен ресурс**.
- Информацията за това как се преминава през двойка ресурси, включително и посоката на преминаване и евентуална информация за поведение при преминаването, се нарича **дъга** (*arc*).

Три типа дъги

за създаване на
споделени връзки
в бази от връзки
(*linkbases*).

за свързване
на бинарни
ресурси

- Дъги, имащи локален начален ресурс и отдалечен краен ресурс, се наричат **изходящи** (*outbound*).
Пример за връзка с такъв тип дъга, имаща за източник локален ресурс, е HTML елементът за котва с маркер <A>;
- Дъги, имащи отдалечен начален ресурс и локален краен ресурс, се наричат **входящи** (*inbound*) – в HTML не е възможно да се дефинират връзки с такива дъги, имащи за цел локален ресурс;
- Дъги, за които нито началният, нито крайният ресурс са локални, се наричат **дъги за трета страна** (*third-party*).
XML XLink, Xpointer и XInclude

Два типа връзки

- **Прости (*simple*)** – притежават само една дъга, с локален начален ресурс и отдалечен краен ресурс. Очевидно тази дъга е от тип изходяща (*outbound*). HTML хипервръзките са прости;
- **Разширени (*extended*)** – могат да свързват произволен брой ресурси, които да бъдат каквато и да е комбинация от отдалечени и локални ресурси. В резултат на това тяхната структура може да бъде доста сложна, включително елементи за посочване на отдалечени ресурси, елементи с локалните ресурси, елементи за определяне на правилата за траверс на дъга, и елементи за определяне на етикети на ресурси и дъги.

Макар и простите връзки да представляват концептуално подклас на разширените връзки, те са с различен синтаксис и не могат лесно да се конвертират до разширени връзки.

Прости XLink връзки 1/2

```
<?xml version="1.0"?>
```

```
<catalog xml:base="http://samples.org/"  
xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<head>
```

```
<title>Virtual Catalog</title>
```

```
</head>
```

```
<body>
```

```
<p>See <myLink xlink:type="simple"  
xlink:href="products.xml">
```

```
Our products</myLink>!</p>
```

```
<li xml:base="/newproducts/">
```

за интерпретиране
на относителните
адреси, зададени в
xlink:href атрибути

Прости XLink връзки 2/2

<item>

<myLink **xlink:type="simple" xlink:href="p1.xml">**

New product 1</myLink>

</item>

<item>

<myLink **xlink:type="simple" xlink:href="p2.xml">**

New product 2</myLink>

</item>

</body>

</catalog>

Важно:

XLink не дефинира елементи, а само атрибути и начина на използването им от съдържащите ги елементи.

Атрибути на връзката 1/3

- **href** – URI на ресурс;
- **type** – глобален атрибут от XLink namespace, задаващ тип на връзката; възможни стойности са: “arc”, “extended”, “locator”, “resource”, “simple”, “title” или “none”;
- **role** – URI на ресурс, описва ролята на елемента;
- **arcrole** – URI на ресурс, описва ролята на връзката;
- **title** – удобно за потребителя заглавие на елемента;
- **show** – описва поведението на представяне на href ресурса при траверс на връзката, т.е. как приложението трябва да представи крайния ресурс на проста връзка (същият атрибут се използва и при дъги) →

Атрибути на връзката 2/3

- **show** – описва поведението на представяне на href ресурса при траверс на връзката, с възможни стойности:
 - **“new”** - съдържанието на документа ще се покаже в нов прозорец, подобно на HTML атрибута TARGET="_blank" на елемента котва;
 - **“replace”** – съдържанието на крайния ресурс ще замени това на началния документ в същия прозорец, подобно на HTML атрибута TARGET="_self" на елемента котва;
 - **“embed”** – съдържанието на крайния ресурс ще се вгради в същия прозорец заедно с това на началния ресурс, както алтернативно съдържание се представя в HTML с атрибута ALT на елемента IMG;
 - **“other”** – няма ограничения върху поведението на приложението при траверс на връзката, но то трябва да потърси друго маркиране в елемента на свързване за определяне на поведение;
 - **“none”** - няма никакви ограничения върху поведението на приложението при траверс на връзката.

Атрибути на връзката 3/3

- **actuate** – описва поведението на съществуване на траверс към **href** ресурс; може да има стойностите:
 - **“onLoad”** – приложението трябва да премине през връзката веднага щом зареди началния ресурс;
 - **“onRequest”** – траверс на връзката се извършва при външно събитие, напр. щракване с мишката върху нея или изтичане на времето за изчакване на редирект;
 - **“other”** и **“none”** - същите както за атрибута **show**.

Елемент, задаващ проста XLink връзка

```
XML Document
<doc ...
xmlns:xlink=".../xlink">
...
<mytag
  xlink:type="simple"
  xlink:role="role_uri"
  xlink:href="ref_uri"
  xlink:show="new"
  xlink:actuate="onLoad">
...
</mytag>
...
</doc>
```

Дефиниция
на роля

Рефериран
ресурс

Инструкция за отваряне на
връзката в нов прозорец при
преминаване през нея

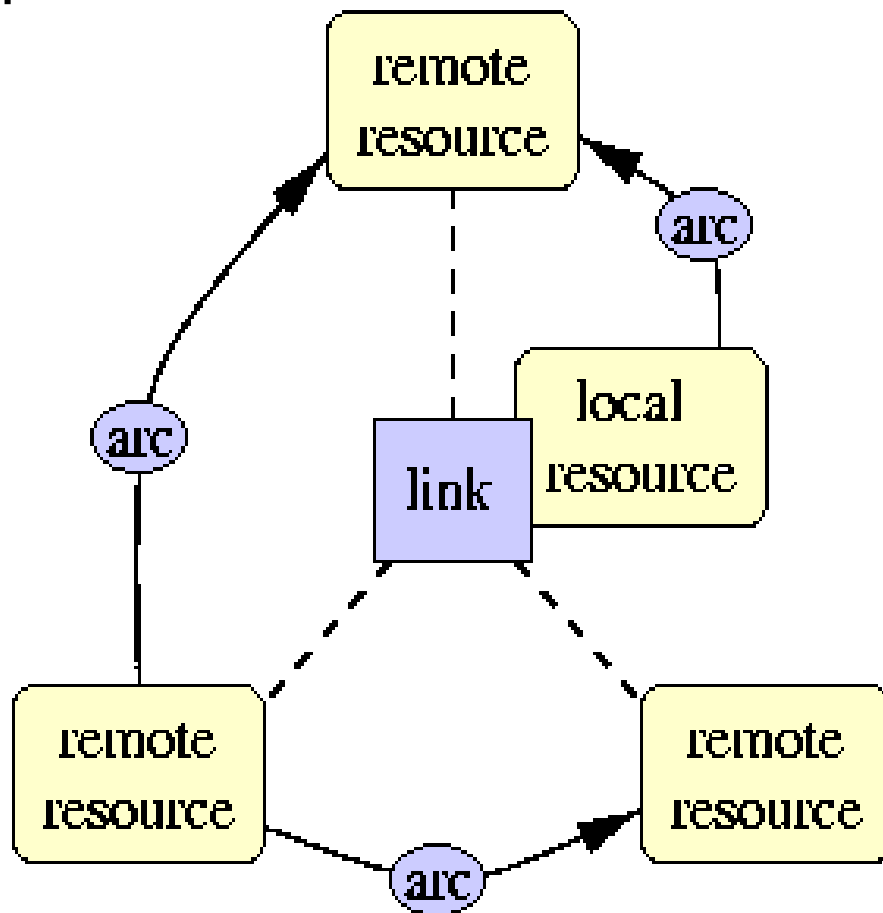
Инструкция за преминаване през
връзката при зареждане на
документа

Разширена XLink връзка 1/2

- асоциира произволен брой ресурси от отдалечен и/или локален тип;
- тя е единственият тип връзка, който може да има входящи дъги;
- разширените XLink връзки може да се съхраняват във външен документ – т. нар. база от връзки (*linkbase*), което води до по-лесна поддръжка и решава проблема с връзките към документи, които не могат да се модифицират (например с бинарно съдържание).

Разширена XLink връзка 2/2

- Разширените връзки могат да съдържат дъги от различен тип;
- Пунктирните линии представят асоциирането на ресурсите с общия свързващ елемент на връзката, който е определен от стандарта Xlink не като име на елемент, а чрез съдържаните от него атрибути.



Елемент с разширена връзка 1/2

Освен атрибути, елементът на разширена връзка в XLink може да съдържа и елементи от следните 4 типа:

1. **локатор** тип елементи, които адресират отдалечени ресурси, участващи в линка - задават се посредством атрибута **type="locator"** и представят отдалечен за връзката ресурс. Те могат да имат и следните атрибути:
 - **href** - URI на ресурса (задължителен)
 - **role** - URI на ресурс, който описва ролята на елемента
 - **title** - описание на елемента за четене от потребители
 - **label** - осигурява етикет, към който елемент от тип дъга може да реферира
2. елементи от тип **ресурс**, които съдържат локалните ресурси, които участват в линка - задават се посредством атрибута **type="resource"** и представят локален за връзката ресурс.

Елемент с разширена връзка 2/2

3. елементи от тип дъга, които осигуряват правила за преминаването между участващите ресурси на връзката - посредством атрибута **type="arc"**, описват връзките между ресурси. Дъгите могат да бъдат изходяща (*outbound*), входяща (*inbound*) и отдалечена (*remote*) – между отдалечени ресурси. Такъв елемент може да ползва атрибутите **show**, **actuate** и **title** със значение, идентично с това атрибутите на елемент, описващ проста връзка. Други възможни атрибути на една дъга може да са:
- **from** - определя етикета на изходните за траверса ресурси
 - **to** - задава етикета за крайните за траверса ресурси
 - **arcrole** - URI на ресурс, който описва ролята на дъгата
4. елементи от тип заглавие - задават се посредством атрибута **type="title"** и дават по-подробно описание от атрибута **title**, напр. при нужда от вложени описания.

Attr.\Type value	simple	extended	locator	arc	resource	title
type	R	R	R	R	R	R
href	O		R			
role	O	O	O		O	
arcrole	O			O		
title	O	O	O	O	O	
show	O			O		
actuate	O			O		
label			O		O	
from				O		
to				O		

XLink атрибути

*Типове на елемент
(по колонии) и
глобални атрибути
за тях (по редове):*

- **required** (**R**)
- **optional** (O)

Пример за разширена връзка

- Следващият пример описва разширена връзка с елемент на свързване между 4 ресурса (три ваканционни почивки и един адрес на физическо лице) и два локатора с адреси на система за банков трансфер през Уеб – един на физическото лице и един на туристически оператор, предлагащ почивките.
- Дъгата **charge_to** задава информация за траверс, реализиращ банков трансфер от сметката на лицето към сметката на оператора (номерата на сметките са указани в URL), а дъгата **deliver_to** задава изпращането на ваучера за почивката.

<?xml version="1.0"?>

<purchase xmlns:xlink="http://www.w3.org/1999/xlink">

<order **xlink:type="extended"** >

<holiday **xlink:type="resource" xlink:label="h1"**>

Carribean Holiday</holiday>

<holiday **xlink:type="resource" xlink:label="h2"**>

Greek Holiday</holiday>

<holiday **xlink:type="resource" xlink:label="h3"**>

Bulgarian Holiday</holiday>

<account **xlink:type="locator" xlink:label="a1"**

xlink:href="http://www.money4nothing.com/cgi/

investbank?bankNo='12-345-67890'"

xlink:title="Ivan Ivanov"/>

<account **xlink:type="locator" xlink:label="b1"**

xlink:href= "http://www.money4nothing.com/cgi/

delawerebank?bankNo='09-876-54321'"

xlink:title="Book Your Travel ASAP"/>

....

<location **xlink:type="resource" xlink:label="d1"**>

<attn>Ivan Ivanov</attn>

<street>123, Dolno blato</street>

<place>Sofia</place>

<zip>1234</zip>

</location >

<charge_to xlink:type=**"arc"** xlink:from=**"a1"** xlink:to=**"b1"**
xlink:title="Ivan pays for holiday by a bank transfer"/>

<deliver_to xlink:type=**"arc"** xlink:from=**"h2"** xlink:to=**"d1"**
xlink:title="Ivan gets holiday voucher at his home"/>

</order>

</purchase>

Друг пример

```
<doc xmlns:xlink=".../xlink">
<order xlink:type="extended" >
  <car xlink:type="resource" xlink:label="c1">Porsche </car>
  <car xlink:type="resource" xlink:label="c2">Ferrari </car>
  <account xlink:type="locator" xlink:label="a1"
    xlink:href="http://www.chargeme.com/cgi/citibank-12-3123"
    xlink:title="Brian Temple"/>
  <location xlink:type="resource" xlink:label="d1">
    <attn>Matthias Hauswirth</attn>
    <street>...</street> ...<zip>80303</zip>
  </location >
  <charge_to xlink:type="arc" xlink:from="c1" xlink:to="a1"
    xlink:title="Brian pays ☹"/>
  <deliver_to xlink:type="arc" xlink:from="c1" xlink:to="d1"
    xlink:title="Matthias gets 😊"/>
</order>
</doc>
```



Bank
Account

none

- Елемент с атрибут *XLink type="none"* няма определено значение.
- Използва се за включване и изключване на *XLink* в отделни елементи.

none пример

DTD

```
...  
<!ELEMENT ref (stuff)>  
<!ATTLIST ref  
  xlink:type (simple|none) "none"  
  xlink:href CDATA #IMPLIED >
```

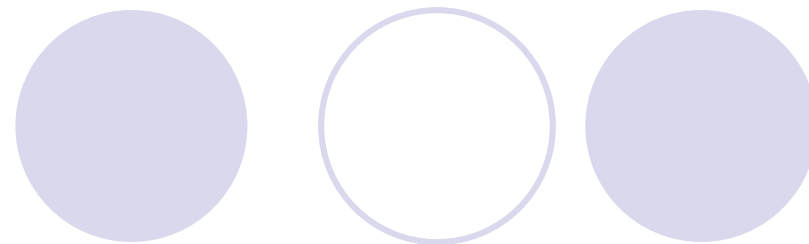
“ref” елемент с връзка

Markup

```
...  
<ref xlink:type="simple" xlink:href="http://x.com">  
  <stuff>stuff</stuff>  
</ref>  
...  
<ref>  
  <stuff>stuff</stuff>  
</ref>
```

“ref” елемент без връзка

Какво е XPointer?



- XPointer е спомагателен език, който не е базиран на XML и служи за определяне на местоположения (*локации*) във вътрешността на XML документи
- Създаден с цел да бъде използван заедно с XLink
- XPointer изразите се добавят в края на URI като идентификатор за фрагмент, за да посочват конкретна част от XML съдържанието, а не целия документ
- XPointer Scheme - W3C Working Draft 19 December 2002
- Спецификация: <http://www.w3.org/TR/xptr-xpointer/>

XPointer като разширение на XPath

- XPointer синтаксисът се основава на синтаксиса на XPath.
- Той използва четирите фундаментални типа данни на XPath (булев, множество от възли, число и низ), като към тях добавя два нови типа: точка (*point*) и обхват или зона (*range*), както и функции за работа с тези типове.
- XPointer добавя също и някои полезни съкращения за често използвани форми на изрази на XPath.
- XPointer представлява разширение на XPath, предназначено за свързване на секции от документ, като задава връзка между XPath изрази и URI връзки.

Проблем на традиционните хипервръзки в HTML

- В HTML документ, връзката се задава към именуван ресурс, напр. елемент-котва или IMAGE;
- за всяка връзка трябва да се поставя котва;
- адресирането е статично - чрез фиксиран идентификатор на фрагмент, което означава, че при промяна на името на фрагмента трябва да се промени и името във връзката;
- могат да бъдат свързани само отделни възли на документа, но не и техни части.

XPointer преодолява тези ограничения

- посредством разширяване на езика XPath, предназначено за употреба на XPath изрази в URI адреси
- чрез езика е възможно свързването да става динамично, на базата на изчисляване на XPath изрази, като се използва адресиране по дефинираните от XPath оси
- това прави тази технология гъвкава и независима от промени в адресирания документ.
- Освен това, езикът предоставя големи възможности за адресиране на части от съдържанието на елемента

Хpointer пример

```
#xpointer(/chapter[3]/elem[@name="foo"])
```

Адресира:

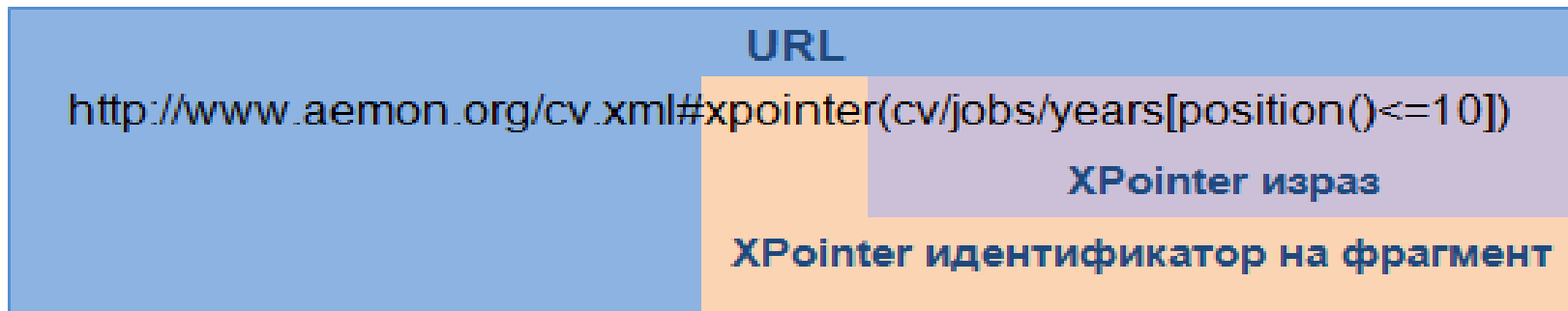
```
<chapter>
```

```
    <elem name="foo"> ...<elem>
```

```
</chapter>
```

Xpointer изрази

- Чрез използването на XPointer в URL се дефинират XML указатели, които се използват за идентифициране на позиции и фрагменти в XML ресурси.
- Това става на база на замяната на опционалния идентификатор за фрагмент с XPointer израз.
- Примерът адресира чрез URL един XML документ, представляващ автобиография, като от документа се извличат първите 10 елемента с име **years**, вложени в йерархията **cv/jobs/years**.



XPointer в браузер?

XPointer би могъл да се ползва за показване на HTML съдържание, стига браузерите да поддържат езика, т.е. да интерпретират XPointer връзки. Тогава при задаване на извеждане на първия елемент с име **product**, намерен в съдържанието на документа, браузерът ще зареди целия документ и ще трябва да покаже въпросния елемент от него.

<a href =

"http://www.sth.org/cat.xml#xpointer(//product[1])">

The name of a product from the catalog.

Засега поддръжката на XPointer в браузери е ограничена.

Важно:

XML

За разлика от традиционното адресиране, чрез XPointer можем да реферираме към повече от един елемента.

XLink, Xpointer и XInclude

37

Последователност от XPointer изрази

- Възможно е също така за бъде зададена последователност от XPointer изрази без задаване на разделители (празните пространства между разделителите са допустими), във вида

URL#xpointer(израз)xpointer(израз)...xpointer(израз)

- При последователност от XPointer изрази, най-напред се изчислява първият израз и ако се върне фрагмент от адресирания документ, то изчислението спира. В противен случай, изразите се изчисляват до връщането на фрагмент или до последния израз.

XPointer израз в XLink връзка

- задаване на XPointer израз в проста XLink връзка към първия срещнат елемент с име **task**, който е наследник на елемента-корен **howto** в документа **manual.xml**:

```
<beforeStarting xlink:type="simple"
```

```
  xlink:href="manual.xml#xpointer(/howto/task[position()=1])">
```

To be read before starting!

```
</beforeStarting>
```

- XPointer допуска използване на пространства от имена в изразите, като за целта префиксът на пространството трябва да се зададе предварително. Например

```
xmlns(html=http://www.w3.org/1999/xhtml) xpointer(//html:div[7])
```

дефинира седмият намерен по ред елемент div от пространството с префикс **html** и адрес равен на **http://www.w3.org/1999/xhtml**.

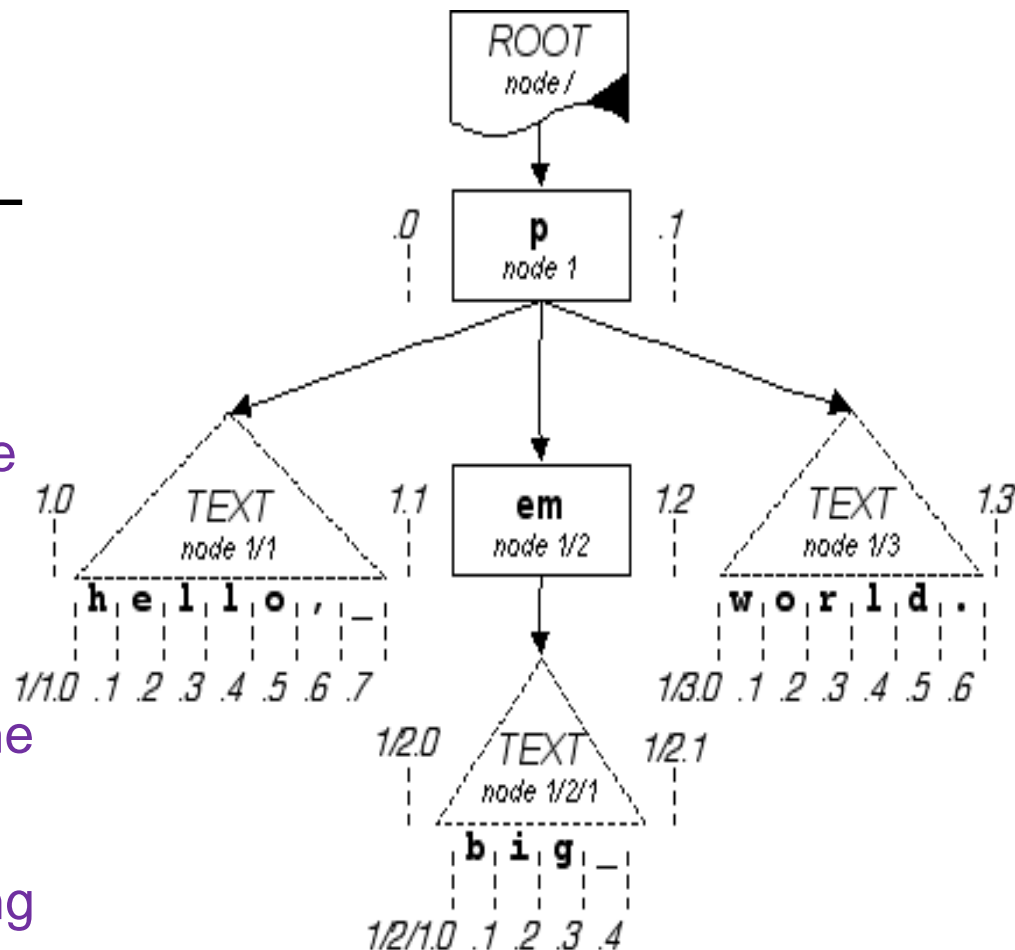
XPointer точка (point) и обхват (range)

точката (*point*) е позиция в XML съдържание

зона или обхват (*range*) - XML съдържанието между двойка крайни точки

Примери:

- point(.0) immediately precedes the root node.
- point(1.0) is just inside the beginning of the first TEXT element.
- point(1.2) is between the end of the em element and the following text node (which contains "world").
- point(1/2/1.1) immediately following the "b" in the middle text node.



XML Inclusions (Xinclude)

- XML Inclusions, или съкратено Xinclude, е универсален механизъм с общо предназначение за изграждане на модулност в XML.
- Спецификацията на XInclude е препоръка на W3C от 2004 г., с втора версия от 2006 г.
- Тя задава XML базиран синтаксис на общ механизъм за сливане на XML документи, посредством задаването на маркери в главния документ за автоматично включване на други документи или части от тях.
- XInclude дава по-добро решение на проблема за конструирането на XML документ от множество документи в сравнение с други подходи като например използването на външни XML единици (external entities).

Проблеми на външните XML единици

- външната XML единица не може да бъде пълноправен, самостоятелен и независим XML документ, понеже в нея не се разрешава задаването на XML декларация или използване на DOCTYPE
- външната XML единица не може да включва други външни единици
- външната XML единица трябва задължително да бъдат добре конструиран XML - това забранява включването на други формати, напр. на код в XML документ
- грешката при зареждане на външна единица е фатална грешка и възстановяването е строго забранено
- може да бъде включена само цялата външна единица, но не и само част от документа
- външните единици трябва да бъдат декларирани в DTD XML, XLink, Xpointer и XInclude документ, което усложнява валидацията

XInclude и XLink

- XLink не уточнява конкретен модел за обработка, а само улеснява откриването на връзки и разпознаването на свързани метаданни от софтуерните приложения.
- За разлика от него, XInclude задава специфично XML в XML преобразуване. Той определя конкретен модел на обработка за обединяване (сливане) на информационни множества.
- XInclude обработка се извършва на най-ниско ниво от XInclude процесор, който подава резултатния за сливането документ към други приложения. При това сливане, XInclude не определя механизъм за DTD валидация на резултата

Синтаксис на Xinclude – езикът задава само два елемента

- Те принадлежат на пространството от имена <http://www.w3.org/2003/XInclude> (с префикс "xi,") и се наричат **include** и **fallback**.
- Спецификацията предоставя XML Schema и DTD, описващи XInclude.

Елемент **xi:include**

Елементът **xi:include** е маркираща инструкция на включване на външен документ, като определя кой документ да се включи и как. Има атрибути:

- **href** - URI референция към документа за включване
- **parse** - може да заема стойностите "xml" (по подразбиране) или "text" в зависимост от това дали се включва XML съдържание или текст без маркиране.
- **xpointer** - опционален XPointer израз, задаващ порция от XML документа за включване; игнорира се при включване на немаркиран текст, т.е. за `parse="text"`
- **encoding** - указва кодирането на включвания документ; при включване на XML съдържание (`parse="xml"`) този атрибут се игнорира
- **accept**, **accept-charset** и **accept-language** - тези атрибути се използват при договарянето на съдържанието в HTTP (*HTTP content negotiation*). Стойностите им се извличат от XInclude процесора и и се задават в хедърите на HTTP заявката с имена Accept, Accept-Charset и Accept-Language. Използват се за случаите, когато един URI връща различни езикови или кодови версии на един и същ ресурс.

Елемент **xi:fallback**

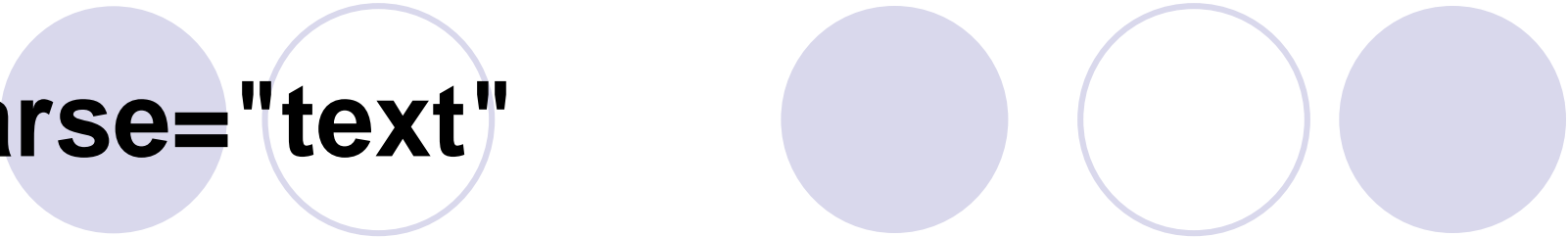
- При сливането на документи нищо не гарантира, че включваните един в друг документи ще са налични по време на обработката от процесора.
- В такъв случай, елементът **xi:fallback** може да се използва като механизъм за възстановяване при липсващ ресурс. Следващият пример показва как се ползва елемента **xi:fallback**. Самият той може да съдържа нов **xi:include** element със свой **xi:fallback** за възстановяване, и т.н.

Пример

```
<page xmlns:xi="http://www.w3.org/2003/XInclude">
  <header>News about the ADOPTA project</header>
  <xi:include href="http://www.adopta.eu/news.xml">
    <xi:fallback>
      <xi:include href="http://62.44.125.62/tmp/news.xml">
        <xi:fallback> Sorry, the ADOPTA server is not
available just now. </xi:fallback>
      </xi:include>
    <xi:fallback>
      </xi:include>
    </xi:include>
  </page>
```

XInclude процес на XML трансформация

- Процесът включва замяната на всеки един `xi:include` елемент с документа, към който той реферира.
- Включването е рекурсивно, като процесорът изгражда дърво на сливанията на база на следване на `xi:include` елементите във всеки нов включван документ до стигането на документ без `xi:include` елементи.
- Поради рекурсивната обработка, цикличните включвания се откриват и третираат като фатални грешки, при които обработването трябва да се спре и цикличното включване да се избегне, напр.:
 - ако елемент **`xi:include`** указва връзка към самия себе си, когато **`parse="xml"`**;
 - ако елемент **`xi:include`** указва връзка към елемент или негов наследник, който вече е бил включен на по-високо ниво.



parse="text"

- При опит да се включват недобре конструирани XML документи се стига до фатална грешка. При **parse="text"** съответният документ се третира като обикновен текст, дори и да е в XML формат, напр.:

```
<doc xmlns:xi="http://www.w3.org/2003/XInclude">
```

A trial with inclusion of XML document as plain text.

```
<xi:include href="sample.xml" parse="text"/>
```

```
</doc>
```

- Това позволява включването на код или на XML съдържание като PCDATA. За целта обаче процесорът заменя началния разделител за маркер с екранната последователност **<**;

Заклучение

- XInclude е W3C стандарт за изграждане на голям XML документите от малки, лесно поддържани XML документи или съдържание с друг формат.
- Ако включваното съдържание е в XML формат, то трябва да е добре оформено (конструирано), но не е задължително да може да се валидира.
- Както XPointer, така и XInclude да се използват заедно с XLinks. И трите езика използват стандарта XML Base за задаване на базов URL за адресираните и включваните документи, съдържащи относителни адреси.
- За съжаление все още тези стандарти не са широко използвани и поддръжката им от много от Уеб браузърите е ограничена.

Допълнителни материали

[CARR] Leslie Carr, Initial Experiences of an XLink Implementation;
<http://journals.ecs.soton.ac.uk/xml4j/xlinkexperience.html>

[JIRAT] Jiri Jirat, XLink Reference;
http://zvon.org/xxl/xlink/Output/xlink_refs.html

[MALER] Eve Maler, XLink and Xpointer Overview; <http://www.oasis-open.org/cover/xlinkMaler980402.html>

[ST LAURENT] Simon St.Laurent, XLinkFilter: An Open Source Java XLink SAX Parser Filter;
<http://www.simonstl.com/projects/xlinkfilter/index.html>

[vdVLIST] Eric van der Vlist, XML Linking Technologies;
<http://www.xml.com/pub/2000/10/04/linking/index.html>

[XMLCOM] XML Resource Guide: XLink;
<http://www.xml.com/pub/Guide/XLink>

