

# **Další XML technologie**

## **XML – teorie a praxe značkovacích jazyků (IZI238)**

Jirka Kosek

Poslední modifikace: \$Date: 2008/05/08 10:30:06 \$

Copyright © 2001-2003 Jiří Kosek

# Obsah

<b>Odkazy</b> .....	<b>3</b>
Odkazy v rámci jednoho dokumentu .....	4
XLink (XML Linking Language) .....	5
XLink (Jednoduché odkazy) .....	6
XLink (Metainformace o odkazu) .....	7
XLink (Rozšířené odkazy) .....	8
XLink (Oblouky (arcs)) .....	9
XML Base .....	10
XPointer (XML Pointer Language) .....	11
Podpora XLinku a XPointeru v aplikacích .....	12
<b>Další XML standardy</b> .....	<b>13</b>
XML Infoset .....	14
XInclude .....	15
XInclude (Podpora v aplikacích) .....	16
Kanonické XML .....	17
XML-Signature .....	18
XML-Encryption .....	21

# Odkazy

Odkazy v rámci jednoho dokumentu .....	4
XLink (XML Linking Language) .....	5
XLink (Jednoduché odkazy) .....	6
XLink (Metainformace o odkazu) .....	7
XLink (Rozšířené odkazy) .....	8
XLink (Oblouky (arcs)) .....	9
XML Base .....	10
XPointer (XML Pointer Language) .....	11
Podpora XLinku a XPointeru v aplikacích .....	12

# Odkazy v rámci jednoho dokumentu

- každý element můžeme jednoznačně označit pomocí atributu typu ID
- každý element může obsahovat atribut ukazující na jiné elementy (typ IDREF nebo IDREFS)
- nově se začíná prosazovat atribut `xml:id`, který nevyžaduje deklaraci typu ve schématu
- hojně se používá v DTD pro tvorbu rozsáhlých dokumentů, které tvoří ucelené celky (např. DocBook pro knihu v jednom XML dokumentu)

## Příklad 1. Ukázka odkazů pomocí ID/IDREF

```
<chapter id="uvod">  
<title>Úvod</title>  
...  
</chapter>
```

```
... <xref linkend="uvod"/> ...
```

## Příklad 2. Totéž s `xml:id`

```
<chapter xml:id="uvod">  
<title>Úvod</title>  
...  
</chapter>
```

```
... <xref linkend="uvod"/> ...
```

# XLink

## XML Linking Language

- mezi dokumenty se musí využít speciální jazyk XLink
  - „zpětná kompatibilita“ s HTML odkazy
  - odkazy mezi více zdroji
  - doplnění odkazů o metadata
  - odkazy lze uložit mimo odkazované dokumenty
- odkaz = explicitní vyjádření vztahu mezi nějakými zdroji
- zdroj = jakýkoliv objekt adresovatelný pomocí URI (soubor, obrázek, výsledek dotazu apod.)
- XPointer – přesné určení místa ve zdroji, případně výběr jeho části
- několik druhů odkazů – rozlišují se pomocí hodnoty atributu `type`
- odkazy se vytvářejí pomocí elementů a atributů patřících do jmenného prostoru <http://www.w3.org/1999/xlink>

# XLink

## Jednoduché odkazy

- obdoba HTML odkazů
- ukázka:

```
<odkaz  
  xmlns:xlink="http://www.w3.org/1999/xlink"  
  xlink:href="http://www.nekde.cz">Odkaz</odkaz>
```

- některé starší implementace XLink vyžadují ještě uvádění  
xlink:type="simple"

# XLink

## Metainformace o odkazu

- `show` – způsob zobrazení (`embed`, `new`, `replace`, `other`, `none`)
- `actuate` – okamžik aktivace odkazu (`onLoad`, `onRequest`, `other`, `none`)
- `title` – název/popis odkazu
- `role` – sémantická role odkazu, identifikuje se pomocí URI adresy

# XLink

## Rozšířené odkazy

- spojují více zdrojů

```
<rozšířený xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="extended">
  <zdroj xlink:href="http://email.seznam.cz"
    xlink:type="locator"
    xlink:title="Seznam"/>
  <zdroj xlink:href="http://mail.atlas.cz"
    xlink:type="locator"
    xlink:title="Atlas"/>
  <zdroj xlink:href="http://www.post.cz"
    xlink:type="locator"
    xlink:title="Post"/>
  <zdroj xlink:href="http://email.centrum.cz"
    xlink:type="locator"
    xlink:title="Centrum"/>
  <popis xlink:type="resource">Přehled
    free poštovních serverů</popis>
</rozšířený>
```

- lokální část (`xlink:type="resource"`) může chybět, odkaz pak spojuje jen externí zdroje
- rozšířený odkaz se může skládat z:
  - `locator` – vzdálený zdroj
  - `resource` – lokální zdroj
  - `title` – popis odkazu
  - `arc` – popis pravidel přechodu mezi jednotlivými zdroji

# XLink

## Oblouky (arcs)

- u rozšířených odkazů není definováno, v jakém směru je lze procházet
- pomocí oblouků to můžeme snadno definovat
- každý zdroj má přiřazené návěští (nemusí být unikátní), oblouky pak popisují směr přechodu pomocí jmen návěští

```
<extendedlink xlink:type="extended">
  <loc xlink:type="locator" xlink:href="..."
        xlink:label="parent" xlink:title="p1" />
  <loc xlink:type="locator" xlink:href="..."
        xlink:label="parent" xlink:title="p2" />
  <loc xlink:type="locator" xlink:href="..."
        xlink:label="child"  xlink:title="c1" />
  <loc xlink:type="locator" xlink:href="..."
        xlink:label="child"  xlink:title="c2" />
  <loc xlink:type="locator" xlink:href="..."
        xlink:label="child"  xlink:title="c3" />
  <go  xlink:type="arc"
        xlink:from="parent"  xlink:to="child" />
</extendedlink>
```

- atribut `arcrole` – sémantická role oblouku, identifikuje se pomocí URI adresy (podobně jako `role` u jednoduchých odkazů)

# XML Base

- napodobení funkčnosti elementu `BASE` z HTML
- určení URI adresy, ke které se vztahují další odkazy v dokumentu
- zapisuje se pomocí speciálního atributu `xml:base`

```
<?xml version="1.0"?>
<doc xml:base="http://example.org/dnes/"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <head>
    <title>Aktuální zprávy</title>
  </head>
  <body>
    <paragraph>Dnešní <link xlink:href="pocasi.xml">předpověď ►
počasí</link>!</paragraph>
    <!-- http://example.org/dnes/pocasi.xml -->
    <olist xml:base="/archiv/">
      <item>
        <link xlink:href="video.xml">video archiv</link>
        <!-- http://example.org/archiv/video.xml -->
      </item>
      <item>
        <link xlink:href="obrazky.xml">archiv obrázků</link>
        <!-- http://example.org/archiv/obrazky.xml -->
      </item>
      <item>
        <link xlink:href="/kontakt.xml">kontaktní informace</link>
        <!-- http://example.org/kontakt.xml -->
      </item>
    </olist>
  </body>
</doc>
```

# XPointer

## XML Pointer Language

- rozšíření XLinku o možnost tvorby odkazů na části dokumentu
- přidává se na konec URL za #

```
http://www.nekde.cz/zdroj.xml#xpointer(...)
```

- existuje několik schémat použitelných s XPointerem – xmlns, element, xpointer
- XPointer výrazy používají rozšířený jazyk XPath
- zachování zpětné kompatibility s HTML odkazy:

```
http://www.nekde.cz/zdroj.xml#kap3  
http://www.nekde.cz/zdroj.xml#xpointer(id('kap3'))
```

- můžeme uvést několik XPointer výrazů, které se vyhodnocují, dokud některý něco nevrátí:

```
element(kap3)xpointer(id('kap3'))xpointer(//*[id='kap3'])
```

- ve schématu element se nepoužívají XPath výrazy, ale pořadová čísla dítěte, která mohou případně začínat ID elementu:

```
element(/1/2/5)
```

- do XPath jsou přidány dva nové typy – rozsah (range) a místo (point) v dokumentu
- několik nových funkcí:
  - `id('kap1')/range-to(id('kap2'))` – vytvoří rozsah pokrývající vše od začátku elementu s ID kap1 do konce elementu s ID kap2
  - `string-range(//title, "Karel IV.")` – vybere všechny texty „Karel IV.“ uvnitř elementů title
  - `covering-range()`, `range-inside()` – funkce ze seznamu uzlů vytvoří rozsah
  - `start-point()`, `end-point()` – zjistí první a poslední místo v daném seznamu uzlů

# Podpora XLinku a XPointeru v aplikacích

- Internet Explorer
  - nepodporuje nic (ani IE6)
  - pomocí XSLT můžeme převést na obyčejné HTML odkazy
  - nelze použít společně s CSS
- Mozilla, Netscape Navigator 6+, Firefox
  - podporují jednoduché XLink odkazy
- XLink a XPointer podporují některé experimentální a jednoúčelové aplikace

# Další XML standardy

XML Infoset .....	14
XInclude .....	15
XInclude (Podpora v aplikacích) .....	16
Kanonické XML .....	17
XML-Signature .....	18
XML-Encryption .....	21

# XML Infoset

- abstraktní datový model pro XML dokumenty
- ke každému XML dokumentu existuje reprezentace v podobě infosetu (stromová reprezentace)
- je využíván v dalších standardech
- PSVI = Post Schema Validation Infoset
  - otypovaný infoset dokument
  - využívá se např. v dotazovacích jazycích (XQuery), které potřebují znát typy dat v jednotlivých elementech a attributech

# XInclude

- umožňuje složení XML dokumentu z několika dílčích souborů
- jednodušší použití než externí textové entity
- umožňuje vkládání XML fragmentů nebo textových souborů
- každý fragment může mít svoje `<!DOCTYPE ...>`

## Příklad 3. Vložení XML souboru

```
<xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
            href="vlozeny_dokument.xml"/>
```

## Příklad 4. Vložení textového souboru

```
<xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
            href="vlozeny_zdrojak_v_javě.java"
            parse="text"
            encoding="windows-1250"/>
```

## Příklad 5. Vložení fragmentu XML

```
<xi:include xmlns:xi="http://www.w3.org/2001/XInclude"
            href="source.xml#xpointer(
            string-range(chapter/p[1], 'Sentence 2')
            /range-to
            (string-range(chapter/p[2]/i, '3.', 1, 11)))/>
```

# XInclude

## Podpora v aplikacích

- libxml2<sup>1</sup>

- velmi dobrá podpora
- vyhodnocení XInclude v dokumentu:

```
xmllint --xinclude dokument.xml
```

- validace jednotlivých částí:

```
xmllint --xinclude --valid dokument.xml
```

v DTD musíme mít deklarované i elementy pro XInclude

- validace „složeného“ dokumentu:

```
xmllint --xinclude --postvalid dokument.xml
```

- Java – standardní součást JDK od 1.5, samostatně v Xercesu
- .NET implementace<sup>2</sup>
- implementace v C++ pro Windows<sup>3</sup>

<sup>1</sup> <http://xmlsoft.org>

<sup>2</sup> <http://mvp-xml.sourceforge.net/xinclude/>

<sup>3</sup> <http://www.xinclude.net/xi/>

# Kanonické XML

- z obsahového hlediska stejný XML dokument může být uložen v souboru několika různými způsoby (nepodstatné mezery, různé konce řádků apod.)
- mnoho aplikací potřebuje XML dokumenty porovnávat (např. digitální podpisy)
- kanonické XML definuje algoritmus pro normalizaci XML dokumentů
- dva dokumenty se považují za identické, pokud je identická jejich kanonizovaná podoba

## Příklad 6. Zdrojový dokument XML

```
<?xml version="1.0" encoding="UTF-8"?>
<a x="&#65;"    mno="pq
  r">
  <b
    >ahoj</b>
</a>
```

## Příklad 7. Výsledek kanonizace

```
<a mno="pq  r" x="A">
  <b>ahoj</b>
</a>
```

# XML-Signature

- standard pro přidání digitálního podpisu k XML dokumentu
- definuje syntaxi a zpracování digitálních podpisů
- podepsat lze libovolná data včetně XML dokumentů
- podporuje mnoho algoritmů a metod
- digitální podpis je zapisován v XML syntaxi
- XML dokument může mít podpis přímo jako svoji část
- lze podepsat jen část dokumentu (výběr pomocí XPath)
- již dnes existují knihovny, které umožňují podepisování a ověřování podpisů v dokumentech
  - <http://www.w3.org/Signature/#Code>

# XML-Signature (Pokračování)

## Příklad 8. Ukázka dokumentu podepsaného pomocí XML Signature

```
<faktura cislo="12/2000" vystaveni="2.2.2000" splatnost="16.2.2000">
  <odberatel>
    <nazev>Poučená, a.s.</nazev>
    <adresa>Široká 21, Praha 1, 110 00</adresa>
    <ico>0987654321</ico>
    <dic>007-0987654321</dic>
  </odberatel>
  <dodavatel>
    <nazev>XMLCompany, s.r.o.</nazev>
    <adresa>Dlouhá 12, Praha 1, 110 00</adresa>
    <ico>1234567890</ico>
    <dic>007-1234567890</dic>
  </dodavatel>
  <polozka>
    <popis>Analýza nasazení XML v podnikovém IS</popis>
    <cena dph="5">50000</cena>
  </polozka>
  <polozka>
    <popis>XML Editor - 10 licencí</popis>
    <cena dph="5">128956</cena>
  </polozka>
  <polozka>
    <popis>Notebook microMini</popis>
    <cena dph="22">89500</cena>
  </polozka>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
  <Reference URI="">
  <Transforms>
  <Transform ►
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>AOz+uJFiTA40hbArhXycZl5V0PM=</DigestValue>
  </Reference>
  </SignedInfo>
  <SignatureValue>IdPPoheBUJEz4dn8cdMlzt2L49ZP0uKm5f0cFe+Fg3AtvGuclQjQA==</SignatureValue>
  <KeyInfo><X509Data><X509Certificate
```

# XML-Signature (Pokračování)

```
>MIIDHjCCAtwCBEOXYIkwCwYHKOZIZjgEAwUAMHUxCzAJBgNVBAYTAmN6MRgwFgYDVQQIEw9DZXNr
YSBSZXB1Ymxpa2ExDzANBgNVBACTB1V0b3BpZTEMMAoGA1UEChMDQklTMREwDwYDVQQLewhWcmF0
bmljZTEaMBGGA1UEAxMRSG9uemEgUG9kZXBpc292YWMwHhcNMDUxMjA3MjIyMjAxWhcNMDYwMzA3
MjIyMjAxWjB1MQswCQYDVQQGEwJjeEYMBYGA1UECBMPQ2Vza2EgUmVwdWJsaWthMQ8wDQYDVQQH
EwZVdG9waWUxDDAKBgNVBAoTA0JJUzERMA8GA1UECXMIVnJhdG5pY2UxGjAYBgNVBAMTEUhvbnpH
IFBvZGVwaXNvdmFjMIIBtzCCASwGByqGSM44BAEwggEfAoGBAP1/U4EddRIpUt9KnC7s5Of2EbdS
PO9EAMMeP4C2USZpRV1AI1H7WT2NWPq/xfW6MPbLm1Vs14E7gB00b/JmYLdrmVC1pJ+f6AR7ECLC
T7up1/63xhv401fnxqimFQ8E+4P208UewwI1VBNaFpEy9nXzrith1yrv8iIDGZ3RSAHHAhUAL2BQ
jxUjC8yykrmCouuEC/BYHPUCgYEA9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVe0
utRZT+ZxBxCBGLRjFnEj6EwoFhO3zwyjMim4TwWeotUfI0o4K0uHiuzpnWRbqN/C/ohNWLx+2J6
ASQ7zKTxvqhRkImog9/hWuWfBpKLZl6Ae1UlZAFMO/7PSSoDgYQAAoGAXnqhIF3rNJZwbO9lhv3p
gjx6CvDAY5z/dpgGZpZvC93deCDWHNffRLsHVPT83GWnvKsElKWwk1M541MEt+e+NHZhMsO/97B5
9Legr7hxBWwIk8wz+nWWdwrX8sgAwFE8BJvCABQYHjc9+50XmsGfQBrXVTfCmY3W3iQB0eqncskw
CwYHKOZIZjgEAwUAAy8AMCwCFGxV1DWTDoop4LNWEtEPxxMhCulAhQ61Zzeb9zxWy+Z0hkRl3Gf
g6nMxA==</X509Certificate></X509Data></KeyInfo></Signature></faktura>
```

# XML-Encryption

- standard pro ukládání zašifrovaných dat do XML
- šifrovat lze:
  - libovolná data
  - XML dokument
  - XML element
  - obsah XML elementu
- podporuje několik algoritmů

## Příklad 9. Ukázka části dokumentu zašifrovaného pomocí XML Encryption

```
<?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>John Smith</Name>
    <CreditCard Limit='5,000' Currency='USD'>
      <Number>
        <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#sifra'
          Type='http://www.w3.org/2001/04/xmlenc#Content'>
          <CipherData>
            <CipherValue>A23B45C56</CipherValue>
          </CipherData>
        </EncryptedData>
      </Number>
      <Issuer>Example Bank</Issuer>
      <Expiration>04/02</Expiration>
    </CreditCard>
  </PaymentInfo>
```