

XML

i nowoczesne metody zarządzania treścią

Wykład 9: XPointer, XInclude i przegląd zastosowań XML-a

Maciej Ogrodniczuk

MIMUW, 5 grudnia 2011

Standard definiuje adresowanie dokumentów XML-owych i ich fragmentów, zgodnie ze składnią URI:

`http://www.sejm.gov.pl/ustawa.xml#def-podatnik`

3 rekomendacje W3C z 2003 r.:

- XPointer Framework
(<http://www.w3.org/TR/xptr-framework/>),
- XPointer element() Scheme
(<http://www.w3.org/TR/xptr-element/>),
- XPointer xmlns() Scheme
(<http://www.w3.org/TR/xptr-xmlns/>),
- XPointer xpointer() Scheme
(<http://www.w3.org/TR/xptr-xpointer/>, Working Draft).

xpointer:

- Ścieżki XPath:

```
http://www.sejm.gov.pl/ustawa.xml  
#xpointer(/art[5]/par[2])
```

xmlns:

- Obsługa przestrzeni nazw
- Do wykorzystania w dalszej części ścieżki:

```
ustawa.xml#xmlns(pr=http://www.sejm.gov.pl/prawo)  
xpointer(/art[5]/par[2])
```

Przykłady:

- element z atrybutem ID o podanej wartości:
`dokument.xml#element(def-podatnik)`
- element o podanej pozycji (ew. względem elementu z atrybutem ID o podanej wartości):
`dokument.xml#element(/1/4/3)`
`dokument.xml#element(def-podatnik/2/3)`
- składnia skrócona:
`dokument.xml#def-podatnik`
`dokument.xml#/1/4/3`
`dokument.xml#def-podatnik/2/3`

Cel: włączanie zawartości jednego dokumentu XML do innego.

Rekomendacja W3C XML Inclusions (XInclude) 1.0 z 2006 r.

Możliwości:

- włączanie całych dokumentów (nazwa pliku lub URL),
- włączanie fragmentów (wskazanych przez XPointer),
- określanie zawartości używanej w razie błędu.

<http://www.w3.org/TR/xinclude/>

Źródło:

```
<wynik>
  <xi:include
    xmlns:xi="http://www.w3.org/2001/XInclude"
    href="salatka.xml#xpointer(/przepis/tytuł)">
    <xi:fallback>
      <błąd>Brak przepisu</błąd>
    </xi:fallback>
  </xi:include>
</wynik>
```

Wynik przetworzenia:

```
<wynik>
  <tytuł>Sałatka z ogórków</tytuł>
</wynik>
```

Możliwości:

- zarządzanie treścią (dokumenty tworzone przez człowieka i przeznaczone dla człowieka, o długim czasie życia), np. Wielka Encyklopedia Powszechna PWN,
- elektroniczna wymiana danych — komunikacja między aplikacjami (dokumenty tworzone oraz przetwarzane automatycznie, zazwyczaj kończące życie wraz z końcem komunikacji), np. komunikaty o błędach,
- wiele, wiele innych!

Zastosowań jest nieskończenie wiele, bo tyle właśnie może powstać XML-owych struktur danych używanych w przeróżnych zastosowaniach.

Po prostu różne modele dokumentów (różne DTD/XML Schema) dla różnych celów:

- zapis tekstu:
 - XML DocBook,
 - TEI – Text Encoding Initiative,
 - ODF (OpenDocument),
 - Office Open XML,
- zapis metainformacji (metadanych):
 - Dublin Core,
 - RDF – Resource Description Framework,
 - Topic Maps,
- zapis informacji naukowych:
 - MathML – Mathematical Markup Language,
- multimedia:
 - SVG – Scalable Vector Graphics.

Przeznaczony do kodowania dokumentacji oprogramowania:

- struktura książki i artykułu,
- dodatkowe struktury specyficzne dla dokumentacji oprogramowania.

Łatwy do dostosowania do własnych potrzeb:

- silnie sparametryzowane DTD (możliwość włączania/wyłączania poszczególnych elementów struktury),
- większość zmian można uzyskać bez modyfikowania DTD.

Prace od 1991 r., obecna wersja: 5.1b4, zarządzana przez OASIS. Przyjęty z otwartymi ramionami przez środowisko open-source i używany do tworzenia dokumentacji wielu projektów (KDE, GNOME, dokumentacja jądra Linuksa, ...)

Z racji wieku standardu istnieje bardzo wiele narzędzi do jego obsługi (edytory WYSIWYG, przekształcenia XSLT, ...)

<http://www.docbook.org>

Element główny <book> (książka), oprócz tego rozdziały, sekcje, streszczenia, wtrącenia, listy...

```
<book xml:id="książka" version="5.0"
      xmlns="http://docbook.org/ns/docbook">
  <title>Świat Zofii</title>
  <chapter xml:id="rozdział1">
    <title>Ogród Edenu</title>
    <para>Komu 3000 lat nie mówią nic, niech w ciemności
      niewiedzy żyje z dnia na dzień.</para>
    <para>--- <emphasis>Goethe</emphasis></para>
  </chapter>
  <chapter xml:id="rozdział2">
    <title>Cylinder</title>
    <para>Zdziwienie jest tą przyczyną,
      ... dla której ludzie zaczęli filozofować.</para>
  </chapter>
</book>
```

TEI (ang. *Text Encoding Initiative*) — standard zapisu danych humanistycznych i lingwistycznych.

- prace od 1987 r.,
- początkowo wersje SGML-owe, obecnie XML-owa,
- obecna wersja: P5 z 2007 r.,
- schemat zawiera moduły (zestawy znaczników) włączane w zależności od potrzeb.

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader> ... </teiHeader>
  <text>
    <body>
      <schemaSpec ident="odd1" start="TEI">
        <moduleRef key="header"/>
        <moduleRef key="core"/>
        <moduleRef key="tei"/>
        <moduleRef key="textstructure"/>
      </schemaSpec>
    </body>
    ...
  </text>
</TEI>
```

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Opowiadania "Złoty żuk" i "Wahadło",
        wersja elektroniczna</title>
      <author>Poe, Edgar Allan</author>
      <respStmt>
        <resp>wyboru dokonał</resp>
        <name>James D. Benson</name>
      </respStmt>
    </titleStmt>
    <publicationStmt> ... </publicationStmt>
    <sourceDesc> ... </sourceDesc>
  </fileDesc>
</teiHeader>
```

```
<teiHeader>
  <fileDesc>
    <titleStmt> ... </titleStmt>
    <publicationStmt>
      <distributor>Oxford Text Archive</distributor>
    </publicationStmt>
    <sourceDesc>
      <bibl>The complete writings of Edgar Allan Poe,
        collected and edited by Phillip S. Foner
        (New York, Citadel Press, 1945)</bibl>
    </sourceDesc>
  </fileDesc>
</teiHeader>
```

MathML to język służący do zapisu wzorów i symboli matematycznych.

Aktualna wersja – 3.0 (wydana w 2010 r.)

Dwa dialekty MathML-a (można łączyć):

- składnia prezentacyjna (ang. *presentational markup*) – 30 elementów, 50 atrybutów,
- składnia semantyczna (ang. *content markup*) – ok. 100 elementów, 12 atrybutów.

<http://www.w3.org/Math/>
<http://www.dessci.com/en/reference/mathml/default.htm>

$$(a + b)^2$$

Składnia prezentacyjna:

```
<mrow>
  <msup>
    <mfenced>
      <mrow>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
      </mrow>
    </mfenced>
    <mn>2</mn>
  </msup>
</mrow>
```

Składnia semantyczna:

```
<mrow>
  <apply>
    <power/>
    <apply>
      <plus/>
      <ci>a</ci>
      <ci>b</ci>
    </apply>
    <cn>2</cn>
  </apply>
</mrow>
```

Pozwala opisywać dwuwymiarową grafikę:

- kształty wektorowe (złożone z linii prostych i krzywych),
- obrazy rastrowe,
- tekst (który jest wciąż tekstem, nie zbiorem kształtów),
- grupowanie, transformacje, filtry, ...

Aktualna wersja: 1.1 z 2011 r. (TR)

<http://www.w3.org/Graphics/SVG/>

Pozwala opisywać dwuwymiarową grafikę:

- kształty wektorowe (złożone z linii prostych i krzywych),
- obrazy rastrowe,
- tekst (który jest wciąż tekstem, nie zbiorem kształtów),
- grupowanie, transformacje, filtry, ...

Aktualna wersja: 1.1 z 2011 r. (TR)

Niebieskie koło z czerwoną obwódką i czarny trójkąt:

```
<svg>
  <circle style="fill: blue; stroke: red"
    cx="200" cy="200" r="100"/>

  </svg>
```

<http://www.w3.org/Graphics/SVG/>

Pozwala opisywać dwuwymiarową grafikę:

- kształty wektorowe (złożone z linii prostych i krzywych),
- obrazy rastrowe,
- tekst (który jest wciąż tekstem, nie zbiorem kształtów),
- grupowanie, transformacje, filtry, ...

Aktualna wersja: 1.1 z 2011 r. (TR)

Niebieskie koło z czerwoną obwódką i czarny trójkąt:

```
<svg>
  <circle style="fill: blue; stroke: red"
    cx="200" cy="200" r="100"/>
  <path d="M250 150 L150 350 L350 350 Z"/>
</svg>
```

<http://www.w3.org/Graphics/SVG/>

Standard reprezentacji metadanych do opisu zasobów (ISO 15836).
Utrzymywany przez DCMI (Dublin Core Metadata Initiative) – organizację promującą stosowanie standardów metadanych i rozwój specjalizowanych słowników pojęć do opisu zasobów.

The original objective of the Dublin Core was to define a set of elements that could be used by authors to describe their own Web resources. (...) The goal was to define a few elements and some simple rules that could be applied by noncatalogers.

<http://dublincore.org>

Standard definiuje 15 elementów:

- `<dc:title>` – tytuł/nazwa opisywanego zasobu,
- `<dc:creator>` – twórca zasobu,
- `<dc:subject>` – temat poruszanego zagadnienia,
- `<dc:description>` – opis,
- `<dc:publisher>` – wydawca (podmiot odpowiedzialny za udostępnienie),
- `<dc:contributor>` – współtwórca,
- `<dc:date>` – data wydarzenia z życia zasobu,
- `<dc:type>` – typ/rodzaj zawartości zasobu,
- `<dc:format>` – format (fizyczny lub cyfrowy),
- `<dc:identifier>` – identyfikator zasobu,
- `<dc:source>` – źródło pochodzenia informacji zawartych w zasobie,
- `<dc:language>` – język zawartości,
- `<dc:relation>` – odniesienie do powiązanych zasobów,
- `<dc:coverage>` – dziedzina/obszar objęty zawartością zasobu,
- `<dc:rights>` – informacja o prawach do zasobu.

RDF:

```
<rdf:Description>
  <dc:title>Wartość energetyczna napojów</dc:title>
  <dc:publisher>Jürgen Lager</dc:publisher>
  <dc:description>Czy lepiej dostać w głowę pustą
    czy pełną butelką piwa?</dc:description>
  <dc:date>2009-06-01</dc:date>
  <dc:language>pl</dc:language>
</rdf:Description>
```

HTML:

```
<meta name="DC.title" lang="pl"
      content="Vaccaonomastyka stosowana">
<meta name="DC.creator" content="Jan Byczewski">
<meta name="DC.description" lang="pl"
      content="Krowy obdarzone imionami dają
      więcej mleka niż bezimienne.">
```

OpenDocument (ODF, OASIS Open Document Format for Office Applications, ISO/IEC 26300) to otwarty standard formatu plików pakietów biurowych.

- zakres: dokumenty tekstowe, arkusze kalkulacyjne, wykresy i prezentacje multimedialne,
- historia:
 - prace od 2002 r.; pierwsza wersja formatu bazowała na XML-owym formacie pakietu OpenOffice.org, obecnie jest rozwijana przez OASIS (Organization for the Advancement of Structured Information Standards),
 - standard ISO od 2006 r.,
 - aktualna wersja standardu: 1.1 (rok 2007), trwają prace nad wersją 1.2,
- cel: stworzenie otwartej alternatywy dla formatów pakietu Microsoft Office.

ODF? Kto tego używa?

Programy obsługujące OpenDocument:

- OpenOffice.org,
- KOffice,
- Google Docs,
- LotusNotes R8...

Prezentowany jako alternatywa dla formatów uchodzących za otwarte (np. Office Open XML), lecz nie powstałych na drodze porozumienia pomiędzy dostawcami oprogramowania biurowego oraz użytkownikami (w tym rządami państw).

Unia Europejska i MSWiA rekomendują OpenDocument jako bazę dla standardowych formatów plików i wymiany dokumentów.

<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf>

Dwa sposoby zapisu dokumentu:

- w pojedynczym dokumencie XML-owym (`<office:document>`),
- w pakiecie zawierającym pliki:
 - `content.xml`: treść dokumentu (`<office:document-content>`),
 - `styles.xml`: style (`<office:document-styles>`),
 - `meta.xml`: metadane (`<office:document-meta>`),
 - `settings.xml`: dane aplikacji (`<office:document-settings>`).

Przestrzenie nazw ODF (czyli zestawy znaczników)

- office – dla elementów ogólnych, nie należących do innych przestrzeni nazw,
- meta – dla metadanych,
- config – dla ustawień aplikacji,
- text – dla elementów opisujących fragmenty tekstu (także np. zawartość komórki arkusza kalkulacyjnego),
- table – dla elementów tabelarycznych,
- drawing,
- presentation,
- form,
- script,
- style,
- ...

```
<office:meta>
  <dc:title>Dlaczego dziękowały nie cierpią
    na wstrząs mózgu?</dc:title>
  <dc:creator>Philip May</dc:creator>
  <dc:date>1999-10-19T15:16:17</dc:date>
  <meta:editing-duration>PT5H10M10S</meta:editing-duration>
  <meta:keyword>Dziękował</meta:keyword>
  <meta:keyword>Wstrząśnienie mózgu</meta:keyword>
  <meta:user-defined meta:name="źródło"
    meta:value-type="string">IgNoble 2006</meta:value>
</meta:user-defined>
</office:meta>
```

```
<office:document-content xmlns:office="..." ...
  office:version="1.0">
  <office:automatic-styles>
    <style:style style:name="P" style:family="paragraph">
      <style:paragraph-properties fo:text-align="center"/>
      <style:text-properties style:font-name="Arial"
        fo:font-size="18pt" fo:font-weight="bold"/>
    </style:style>
  </office:automatic-styles>
  <office:body>
    <office:text>
      <text:p text:style-name="P">Format OpenDocument
        jest standardem <text:a xlink:type="simple"
          xlink:href="http://oasis-open.org">
          OASIS</text:a></text:p>
    </office:text>
  </office:body>
```

Office Open XML (OOXML, OpenXML, MSOOXML) to alternatywna do OpenDocument specyfikacja dokumentów elektronicznych mająca zapewnić wsteczną zgodność z istniejącymi dokumentami w formatach Microsoft Office oraz pełną obsługę funkcji oferowanych przez ten pakiet.

Historia:

- 2006: specyfikacja zatwierdzona jako standard ECMA International,
- 2007: porażka w głosowaniu nad szybką ścieżką zatwierdzenia OOXML jako standardu ISO,
- 2008: standard jednak przyjęty.

<http://www.ecma-international.org/publications/standards/Ecma-376.htm>

Struktura pliku:

- dokument OOXML to pakiet (skompresowany ZIP),
- wewnątrz: pliki XML-owe zawierające dane, metadane, ... zgodne z językami znacznikowymi:
 - WordprocessingML (`\word\document.xml`, `\word\styles.xml...`),
 - SpreadsheetML,
 - PresentationML,
 - ...
- kompatybilne z binarnymi formatami MS Office w zakresie reprezentowalności ich możliwości,
- niekompatybilne z XML-owymi formatami Office XP i 2003.

Typowa struktura dokumentu:

- `<w:document>` – element główny,
- `<w:body>` – kolekcja akapitów i własności sekcji,
- `<w:p>` – akapit (kolekcja fragmentów),
- `<w:r>` – fragment tekstu o określonych własnościach (ciąg, ang. *run*),
- `<w:t>` – czysty tekst (ang. *text range*).

```
<w:document xmlns:w="...">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello, world.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

```
<w:p>
  <w:pPr>
    <w:jc w:val="center"/>
  </w:pPr>
  <w:r>
    <w:t>To</w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:b/>
    </w:rPr>
    <w:t>naprawdę</w:t>
  </w:r>
  <w:r>
    <w:t>Ty?</w:t>
  </w:r>
</w:p>
```

Fanatycy ODF:

Wasza specyfikacja ma 6000 stron! Ciężko będzie to wdrożyć w całości...

Fanatycy ODF:

Wasza specyfikacja ma 6000 stron! Ciężko będzie to wdrożyć w całości...

Wielbiciel OOXML:

Ale tam są przykłady i dokładne opisy, a w ODF-ie tylko spis znaczników!

Poza tym musimy zachować kompatybilność z dokumentami Office, bo jest ich dużo, czego nie można powiedzieć o ODF...

Fanatycy ODF:

Wasza specyfikacja ma 6000 stron! Ciężko będzie to wdrożyć w całości...

Wielbiciele OOXML:

Ale tam są przykłady i dokładne opisy, a w ODF-ie tylko spis znaczników!

Poza tym musimy zachować kompatybilność z dokumentami Office, bo jest ich dużo, czego nie można powiedzieć o ODF...

Fanatycy ODF:

W ODF-ie można reprezentować to samo, co w OOXML-u. Więcej i lepiej, bo wasza specyfikacja zawiera błędy. Dlaczego rok 1900 jest przestępny?

Wielbiciele OOXML:

A ODF jest prostacki w porównaniu z OOXML-em! Używacie jednego typu tabeli dla wszystkich rodzajów dokumentów...

Wielbiciele OOXML:

A ODF jest prostacki w porównaniu z OOXML-em! Używacie jednego typu tabeli dla wszystkich rodzajów dokumentów...

Fanatycy ODF:

Nie jest prostacki, tylko prosty. I elegancki w swej prostocie. Gdybyście nam pomogli, też byście tak mieli. MS był członkiem OASIS, kiedy powstawał ODF. Dlaczego robiliście swoje pod stołem?

OOXML kontra OpenDocument: wojna standardów

Wielbiciele OOXML:

A ODF jest prostacki w porównaniu z OOXML-em! Używacie jednego typu tabeli dla wszystkich rodzajów dokumentów...

Fanatycy ODF:

Nie jest prostacki, tylko prosty. I elegancki w swej prostocie. Gdybyście nam pomogli, też byście tak mieli. MS był członkiem OASIS, kiedy powstawał ODF. Dlaczego robiliście swoje pod stołem?

Wielbiciele OOXML:

Bo chcieliście dać przewagę OpenOffice. Nie dalibyśmy rady tego tego wdrożyć — w 2005 r. OpenOffice już używał ODF-u, a my dopiero zaczynaliśmy.

Fanatycy ODF:

Eeee tam, przecież pracowaliśmy nad tym już od 2002 r. Poza tym byliśmy pierwsi, a skoro jest już jeden standard, to dlaczego go nie wykorzystać? Wy w ogóle nie lubicie standardów, bo ODF używa MathML, SVG, a wy wszystko po swojemu, nawet kody języków macie inne niż cały świat... A co z prawami autorskimi i patentowymi?

Fanatycy ODF:

Eeee tam, przecież pracowaliśmy nad tym już od 2002 r. Poza tym byliśmy pierwsi, a skoro jest już jeden standard, to dlaczego go nie wykorzystać? Wy w ogóle nie lubicie standardów, bo ODF używa MathML, SVG, a wy wszystko po swojemu, nawet kody języków macie inne niż cały świat... A co z prawami autorskimi i patentowymi?

Wielbiciele OOXML:

Grrrrr!!!

Fanatycy ODF:

Eeee tam, przecież pracowaliśmy nad tym już od 2002 r. Poza tym byliśmy pierwsi, a skoro jest już jeden standard, to dlaczego go nie wykorzystać? Wy w ogóle nie lubicie standardów, bo ODF używa MathML, SVG, a wy wszystko po swojemu, nawet kody języków macie inne niż cały świat... A co z prawami autorskimi i patentowymi?

Wielbiciele OOXML:

Grrrrr!!!

Fanatycy ODF:

Microsoft isn't the answer.

Microsoft is the question, and the answer is NO.

Odnośniki HTML-owe:

- łączą dwa dokumenty: źródło i cel linku,
- źródłem linku jest zawsze element opisujący link (<A>,).

XLink — rozszerzona koncepcja powiązań:

- informacja o linkach zapisana w dowolnym elemencie:
 - nieistotna nazwa elementu,
 - istotne atrybuty z przestrzeni nazw XLink,
- więcej niż dwa końce linku (hiperlink → relacja),
- możliwość reprezentacji linku poza wiązanyymi zasobami.

Status:

- korzenie historyczne: HyTime,
- XLink 1.0 – rekomendacja W3C, czerwiec 2001,
- XLink 1.1 – wersja robocza (aktualizacja w 2008).

Zasób (ang. *resource*) — dowolna adresowalna jednostka informacji lub usługa (plik, program, wynik kwerendy).

Dowiązanie (ang. *link*) – wyrażona jawnie, przy pomocy **elementu wiążącego** (ang. *linking element*), relacja pomiędzy zasobami **uczestniczącymi** (ang. *participate*) w dowiązaniu.

Przejsście (ang. *traversal*) – użycie pary zasobów (źródłowego – *starting* i docelowego – *ending*) połączonej dowiązaniem.

Łuk (ang. *arc*) – informacja o przejściu między zasobami oznaczonymi dwoma etykietami (w określonym kierunku):

- **łuk wychodzący** (ang. *outbound*) – z zasobu lokalnego do zewnętrznego,
- **łuk wchodzący** (ang. *inbound*) – z zewnętrznego do lokalnego,
- **łuk niezależny** (ang. *third party*) – między zewnętrznymi.

UWAGA: zasób jest traktowany jako **zewnętrzny** (ang. *remote*) zawsze, gdy jest adresowany za pomocą URI (nawet, gdy znajduje się w tym samym dokumencie czy elemencie linkującym co posługujący się nim link.

Link prosty (ang. *simple link*):

- to link wychodzący,
- wiąże dokładnie dwa zasoby: lokalny z zewnętrznym,
- zawiera dokładnie jeden łuk pomiędzy zasobami.

Link rozszerzony (ang. *extended link*):

- wiąże dowolną liczbę zasobów lokalnych i zewnętrznych,
- łuki opisują sposoby przechodzenia pomiędzy zasobami,
- określa role zasobów uczestniczących w linku,
- określa role łuków.

```
<artykuł xmlns:xlink="http://www.w3.org/1999/xlink">
  <autor>Diego A. Golombek</autor>
  <tytuł>Pozytywny wpływ <sponsor xlink:type="simple"
    xlink:href="http://www.example.com/
    shop.xml#viagra">viagry</sponsor>
    na neutralizowanie zespołu nagłej zmiany
    strefy czasowej u chomików</tytuł>
</artykuł>
```

Przejdźcie od każdego z rodziców do każdego dziecka:

```
<rodzina xlink:type="extended"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <osoba xlink:type="locator" xlink:href="hipcio.xml"
    xlink:label="rodzic" xlink:title="Hipolit"/>
  <osoba xlink:type="locator" xlink:href="wiesia.xml"
    xlink:label="rodzic" xlink:title="Wiesława"/>
  <osoba xlink:type="locator" xlink:href="roxy.xml"
    xlink:label="dziecko" xlink:title="Roksana"/>
  <osoba xlink:type="locator" xlink:href="oli.xml"
    xlink:label="dziecko" xlink:title="Oliwier"/>
  <osoba xlink:type="locator" xlink:href="nika.xml"
    xlink:label="dziecko" xlink:title="Nicola"/>
  <link xlink:type="arc" xlink:from="rodzic"
    xlink:to="dziecko"/>
</rodzina>
```

Atrybuty w linkach rozszerzonych

<code>type</code>	rola elementu w linku (<code>simple</code> , <code>extended</code> , <code>locator</code> , <code>arc</code> , <code>resource</code> , <code>title</code> , <code>none</code>)
<code>href</code>	URI zasobu zewnętrznego
<code>role</code>	abstrakcyjny identyfikator roli zasobu w powiązaniu (URI)
<code>arcrole</code>	j.w., ale dla pojedynczego łuku między zasobami
<code>title</code>	etykieta tekstowa zasobu, łuku (opcjonalna)
<code>show</code>	jak prezentować (<code>new</code> , <code>replace</code> , <code>embed</code> , <code>other</code> , <code>none</code>)
<code>actuate</code>	kiedy aktywować (<code>onLoad</code> , <code>onRequest</code> , <code>other</code> , <code>none</code>)
<code>label</code>	etykieta zasobu używana jako identyfikator (do użycia <code>we from</code> i <code>to</code> , nie muszą być unikalne)
<code>from</code> , <code>to</code>	wskazanie (w łuku) na etykietę zasobu

Link prosty:

```
<link xlink:href="...">Jam Łasica</link>
```

Link rozszerzony:

```
<link xlink:type="extended">  
  <resource xlink:type="resource"  
    xlink:label="local">Jam Łasica</resource>  
  <locator xlink:type="locator" xlink:href="..."  
    xlink:label="remote" xlink:role="..."  
    xlink:title="..."/>  
  <go xlink:type="arc" xlink:from="local"  
    xlink:to="remote" xlink:arcrole="..."  
    xlink:show="replace" xlink:actuate="onRequest"/>  
</link>
```

Zastosowania:

- organizowanie, kojarzenie zasobów, nawet gdy nie mamy prawa zapisu,
- dostarczanie wartości dodanej – zbiorów linków.

Zasięg:

- lokalny – serwery linków operujące na bazie linków,
- Internet?

Problemy:

- wizualizacja linków rozszerzonych,
- synchronizacja zasobów i linków (Internet).

Standard powstał w IBM (2001), obecnie opiekę nad nim przejęła organizacja OASIS (od 2005 r.)

Główne pojęcia:

- informacja zgromadzona w **tematach** (ang. *topics*) – porcjach informacji o danym zagadnieniu (tytuł, tekst, obrazki),
- **dokumenty-mapy** (ang. *ditamaps*) są uporządkowanym zbiorem **odwołań do tematów** (ang. *topicrefs*), tworzonym z uwzględnieniem wartości metadanych tematów,
- **odwołania śródtekstowe** (ang. *conrefs*) umożliwiają włączanie komponentów do treści tematów.

<http://www.ditaworld.com>

Specjalizacja: tematy mogą być hierarchicznie doszczegóławiane (w określonym, zdefiniowanym w specyfikacji formalizmie); sama specyfikacja wprowadza trzy zawężenia „pierwotne”:

- **zadanie** (ang. *task*) – procedura opisująca sposób wykonania danej czynności, złożona z listy kroków,
- **pojęcie** (ang. *concept*) – wyjaśnienie procesów wymaganych do zrozumienia kroków zadania,
- **materiał zależny** (ang. *reference*).

Efekt:

- dokumenty specjalizowane pozostają dokumentami typu bazowego,
- wiążemy przetwarzanie typu zależnego z przetwarzaniem typu bazowego,
- zachowujemy możliwość rozszerzenia łańcucha przetwarzania o nowe elementy.

```
<task id="maintaindatabase" xml:lang="pl">
  <title>Administracja bazą danych</title>
  <shortdesc>Administracja bazą jest łatwa.</shortdesc>
  <taskbody>
    <steps-unordered>
      <step>
        <cmd>Jeśli baza nie działa, zrestartuj ją.</cmd>
      </step>
      <step>
        <cmd>Jeśli baza działa wolno, namów użytkowników,
          by używali jej oszczędnie.</cmd>
      </step>
    </steps-unordered>
  </taskbody>
</task>
```