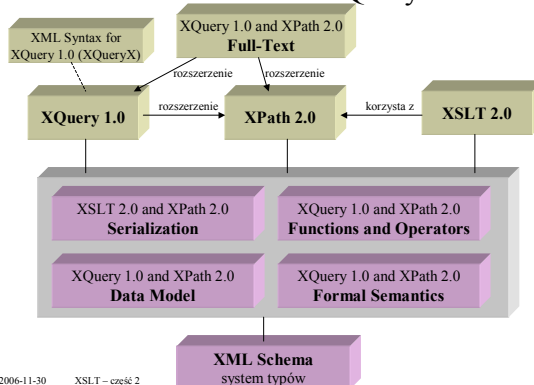


## Inne standardy związane z XML-em

## XPath 2.0 / XSLT 2.0 / XQuery 1.0



## XPath 2.0

- Język:
  - już nie tylko do wskazywania węzłów w dokumencie,
  - operujący na sekwencjach (wartości atomowych lub węzłów),
  - funkcyjny język programowania.
- Model danych:
  - wykorzystanie typów danych (19 typów pierwotnych XML Schema),
  - wykorzystanie informacji dostępnych w Post-Schema Validation Infoset (typów danych, wartości domyślnych, itp.).

2006-11-30 XSLT – część 2

3

## XPath 2.0 – funkcyjny język programowania

- Operacje na sekwencjach, np.:  
`sum(for $x in /order/item return $x/price * $x/quantity)`  
`$x intersect /order/item/number`
- Wyrażenia warunkowe, np.:  
`if ($widget1/unit-cost < $widget2/unit-cost)`  
`then $widget1`  
`else $widget2`
- Kwantyfikatory, np.:  
`some $x in /students/student/name satisfies $x = "Fred"`  
`every $x in /students/student/name satisfies $x = "Fred"`

Przykłady na podstawie:  
 Lenz, E., *What's New in XPath 2.0*, <http://www.xml.com/pub/a/2002/03/20/xpath2.html>

2006-11-30 XSLT – część 2

4

## XSLT 2.0

- Najważniejsze zmiany w stosunku do XSLT 1.0:
  - możliwość wykorzystania typów ze schematu XML Schema,
  - wykorzystanie XPath 2.0,
  - grupowanie węzłów (`for-each-group`),
  - definiowanie i wywołanie własnych funkcji,
  - generowanie wielu dokumentów wyjściowych,
  - koniec z fragmentami drzewa wynikowego (teraz używamy sekwencji XPath 2.0).

2006-11-30 XSLT – część 2

5

## XSLT 2.0 – wiele dokumentów wyjściowych

```
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml"
  <xsl:output method="xhtml"/>
  <xsl:output method="text" name="textFormat"/>

  <xsl:template match="/">
    <html><body>
      <p>Here is a list of links to text files:</p>
      <ul><xsl:apply-templates select="//textBlob"/>
      </ul>
    </body></html>
  </xsl:template>

  <xsl:template match="textBlob">
    <xsl:variable name="uri"
      select="concat('text', position(), '.txt')"/>
    <li><a href="{uri}"><xsl:value-of select="{uri}"></a></li>
    <xsl:result-document href="{uri}" format="textFormat">
      <xsl:value-of select="."/>
    </xsl:result-document>
  </xsl:template>
</xsl:stylesheet>
```

Źródło: Lenz, E., *What's New in XSLT 2.0*, <http://www.xml.com/pub/a/2002/04/10/xslt2.html>

2006-11-30 XSLT – część 2

6

## XSLT 2.0 – grupowanie

```

<cities>
  <city name="milan" country="italy" pop="5"/>
  <city name="paris" country="france" pop="7"/>
  <city name="munich" country="germany" pop="4"/>
  <city name="lyon" country="france" pop="2"/>
  <city name="venice" country="italy" pop="1"/>
</cities>

<xsl:for-each-group select="cities/city"
  group-by="@country">
  <tr>
    <td><xsl:value-of select="@country"/></td>
    <td><xsl:value-of select="current-group()/@name"
      separator=", "/></td>
    <td><xsl:value-of select="sum(current-group()/@pop)"/>
    </td>
  </tr>
</xsl:for-each-group>

```

Źródło: Lenz, E., *What's New in XSLT 2.0*, <http://www.xml.com/pub/a/2002/04/10/xsh2.html>

2006-11-30 XSLT – część 2

7

## XQuery – zapytania o dokumenty XML

- Język zapytań o dokumenty XML:
  - podobny do SQL-a,
  - pozwala pytać o kolekcje dokumentów,
  - rozszerza funkcjonalność XPath m. in. o:
    - zmienne,
    - wyrażenia FLWOR (for, let, where, order by, return),
    - złączenia,
    - wyrażenia warunkowe,
    - definiowanie funkcji,
    - konstruktory, umożliwiające tworzenie struktur XML w zapytaniu,
    - wyrażenia walidujące względem schematu.
- Dostępne składnie:
  - SQL-opodobna,
  - oparta na XML-u.

2006-11-30 XSLT – część 2

8

## XQuery – przykład

- Zwróć nazwę każdego wydawcy i średnią cenę jego książek:
 

```

FOR $p IN distinct(document("bib.xml"))//publisher
LET $a :=
  avg(document("bib.xml")//book[publisher = $p]/price)
RETURN
  <publisher>
    <name>{ $p/text() }</name>
    <avgprice>{ $a }</avgprice>
  </publisher>

```

Źródło: *XML Syntax for XQuery 1.0 (XQueryX)*, <http://www.w3.org/TR/xqueryx>

2006-11-30 XSLT – część 2

9

## Who is who?

- XPath 2.0:
  - *The primary purpose of XPath is to address the nodes of XML or trees.*
  - *XPath 2.0 is an expression language that allows the processing of values conforming to the data model.*
  - *XPath is designed to be embedded in a host language such as XSLT 2.0 or XQuery.*
  - *XPath has a natural subset that can be used for matching (testing whether or not a node matches a pattern).*
- XSLT 2.0:
  - *A language for transforming XML documents into other XML documents.*
- XQuery 1.0:
  - *A language for retrieving and interpreting information from a broad spectrum of XML information sources, including both databases and documents.*

Cytaty z *Candidate Recommendations*.

2006-11-30 XSLT – część 2

10

## XPointer – adresowanie fragmentów dokumentu XML

- XPath „opakowany” w składnię URI, np.:
 

```

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

```
- Dodatkowe możliwości:
  - proste adresowanie elementów opatrzonych atrybutami ID, np.:
 

```

urlopy

```
  - adresowanie elementów bez znajomości struktury, z wykorzystaniem atrybutów ID jako kotwic, np.:
 

```

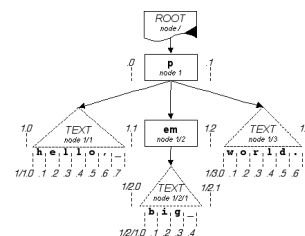
element(/4/2/3)
element(urlopy/2/3)

```
  - wskazanie punktu w dokumentcie,
  - wskazanie zakresu pomiędzy dwoma punktami,
  - wskazanie punktów w tekście i fragmentów tekstu.

2006-11-30 XSLT – część 2

11

## XPointer – punkty



- Użycie:
 

```

point(1/2/1.1)

```

Źródło: *XPointer (xpointer) Scheme*, W3C Working Draft 19 December 2002, <http://www.w3.org/TR/xptr-xpointer/>

2006-11-30 XSLT – część 2

12

## XPointer – status

- Rekomendacje W3C z 25 marca 2003:
  - XPointer Framework,
  - XPointer element() scheme,
  - XPointer xmlns() scheme.
- W3C Working Draft:
  - XPointer xpointer() scheme.

## XInclude – łączenie zawartości dokumentów

- Załączanie zawartości jednego dokumentu XML do innego:
  - pozwala załączyć fragment dokumentu (wskazany XPointer-em),
  - pozwala określić zawartość używaną w razie błędu.
- Przykład:

```
<file name="salatka.xml">
  <xi:include
    xmlns:xi="http://www.w3.org/2001/XInclude"
    href="salatka.xml#xpointer (/przepis/tytul)"/>
</file>
```

  - po przetworzeniu procesorem XInclude:

```
<file name="salatka.xml">
  <tytul>Sałatka z ogórków</tytul>
</file>
```
- Status: Rekomendacja W3C z 20 grudnia 2004.

## XLink – dowiązania w XML-u

- Linki jakiego znamy (HTML):
  - łączy dwa dokumenty: źródło i cel linku,
  - źródłem linku jest zawsze element opisujący link (A, IMG).
- XLink – rozszerzona koncepcja dowiązań:
  - dowolne elementy przechowują informacje o linkach,
  - informacja o linkach poza połączone dokumentami,
  - więcej niż dwa końce linku.
- Status:
  - rekomendacja W3C z 27 czerwca 2001,
  - korzenie historyczne: HyTime.

## Terminologia

- **Zasób (resource)** – dowolna adresowalna jednostka informacji lub usługi.
- **Dowiązanie (link)** – jawnie wyrażona (przy pomocy **elementu wiążącego (linking element)**) relacja pomiędzy zasobami.
  - te zasoby **uczestniczą (participate)** w dowiązaniu.
- **Przejsięcie (traversal)** – użycie pary zasobów połączonej dowiązaniem.
- **Łuk (arc)** – informacja o przejściu między dwoma zasobami (kierunek, zachowanie aplikacji, itp.):
  - wychodzący (*outbound*),
  - wchodzący (*inbound*),
  - niezależny (*third party*).

## Dowiązania XLink

- **Extended link:**
  - wiąże dowolną liczbę zasobów:
    - zasoby zewnętrzne (np. odwołania do innych dokumentów),
    - zasoby lokalne (zawarte w elemencie wiążącym).
  - łuki opisujące sposoby przechodzenia pomiędzy zasobami,
  - role zasobów uczestniczących w linku,
  - role łuków.
- **Simple link:**
  - link wychodzący,
  - wiąże dokładnie 2 zasoby: 1 lokalny i 1 zewnętrzny,
  - jeden łuk z zasobu lokalnego do zewnętrznego.

## Simple link – przykład

```
<osoba xmlns:xlink="http://www.w3.org/1999/xlink">
  <nazwisko>Kopernik, Mikołaj</nazwisko>
  <biogram>Wybitny polski astronom, matematyk, lekarz,
  prawnik, tłumacz poezji włoskiej i ekonomista,
  pochodził z rodziny wywodzącej się z mieszczan
  krakowskich. Urodzony w <geogr xlink:type="simple"
  xlink:href="Torun.xml">Toruniu</geogr>.</biogram>
</osoba>
```

## Extended link - przykład

```
<fikcja xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:type="extended">
  <wypowiedz xlink:type="resource">Kopernik była
  kobietą!</wypowiedz>
  <film xlink:type="locator" xlink:href="seksmisja.xml"
  xlink:title="Seksmisja"/>
  <osoba xlink:type="locator" xlink:href="kopernik.xml"
  xlink:title="Kopernik, Mikołaj"/>
  <pojecie xlink:type="locator"
  xlink:href="kobieta.xml"
  xlink:title="kobieta"/>
</fikcja>
```

2006-11-30 XSLT - część 2

19

## Przyszłość XLink

- 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 *extended links*,
  - synchronizacja zasobów i linków (Internet).

2006-11-30 XSLT - część 2

20

## XForms - zaawansowane formularze

- Odpowiedź na ograniczenia formularzy w HTML-u:
  - kontrola dziedziny wprowadzanych danych po stronie klienta,
  - specyfikowanie pól obowiązkowych i opcjonalnych,
  - zależności między polami, np.:
    - wartość pola jest funkcją wartości innych pól;
    - pole jest widoczne tylko przy określonej zawartości innych pól;
  - wyzwalacze aktywowane zdarzeniami interfejsu użytkownika,
  - bogaty zasób kontroltek, np.:
    - powtórzenia, grupy, wielopoziomowe wybory,
    - suwaki do wyboru wartości z zakresu,
    - kontrolki wyboru plików.
- Status:
  - rekomendacja W3C z 14 października 2003 r.,
  - możliwość zanurzania w XHTML-u.

2006-11-30 XSLT - część 2

21

## XForms - prosty formularz

```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xf="http://www.w3.org/2002/xforms">
<head>
  <xf:model id="modell">
    <xf:instance>
      <dane xmlns=""><imie/><nazwisko/></dane>
    </xf:instance>
    <xf:submission id="submit" method="post"
      action="http://xformstest.org/test"/>
  </xf:model>
</head>
<body>
  <h1>Wpisz swoje dane</h1>
  <xf:input model="modell" ref="imie">
    <xf:label>Imię</xf:label>
  </xf:input>
  <xf:input model="modell" ref="nazwisko">
    <xf:label>Nazwisko</xf:label>
  </xf:input>
  <xf:submit submission="submit">
    <xf:label>Save</xf:label>
  </xf:submit>
</body>
</html>
```

2006-11-30 XSLT - część 2

22

## XForms - przykład (program X-Smiles)

Wpisz swoje zamówienie

Dział: Kasa | Aktualna liczba paragonów: 2

Karta: 1234567890 | Data wazności: 30 maj 2003

Netto: 500  
VAT: 110  
Do zapłaty: 610

Zamawiam!

Źródło: Karzenczo, P., Co tam panie w XML-u?, Software 2.0, 6/2003

2006-11-30 XSLT - część 2

23

## Gdzie szukać dalej

- DuCharme, B., *The Path of Control*  
[www.xml.com/pub/a/2005/05/04/tr-xml.html](http://www.xml.com/pub/a/2005/05/04/tr-xml.html)
- Arciniegas, A. F., *What is XLink?*  
[www.xml.com/pub/a/2000/09/xlink](http://www.xml.com/pub/a/2000/09/xlink)
- Carr, L., *Initial Experiences of an XLink Implementation*  
[journals.ecs.soton.ac.uk/xml4j/xlinkexperience.html](http://journals.ecs.soton.ac.uk/xml4j/xlinkexperience.html)
- *XPointer tutorial*  
[www.zvon.org/xxl/xpointer/tutorial/OutputExamples/xpointer\\_tut.html](http://www.zvon.org/xxl/xpointer/tutorial/OutputExamples/xpointer_tut.html)
- *Tamino XQuery Demo*  
[tamino.demzone.softwareag.com/demoXQuery](http://tamino.demzone.softwareag.com/demoXQuery)



2006-11-30 XSLT - część 2

24

## Gdzie szukać dalej

- XForms Institute. Interactive XForms School  
🌐 [xformsinstitute.com](http://xformsinstitute.com)
- Dubinko, M., *What Are XForms?*  
🌐 [www.xml.com/pub/a/2001/09/05/xforms.html](http://www.xml.com/pub/a/2001/09/05/xforms.html)
- Dubinko, M., *Ten Favorite XForms Engines*  
🌐 [www.xml.com/pub/a/2005/02/09/xforms.html](http://www.xml.com/pub/a/2005/02/09/xforms.html)
- Kazienko, P., *Co tam panie w XML-u?*  
☐ Software 2.0, nr 6/2003, Wydawnictwo Software

