

Programowanie komunikacji człowiek-komputer

dr inż. Joanna Ochelska-Mierzejewska

XPath

Język ścieżek XML, język ścieżek rozszerzalnego języka znaczników

XPath

- Języki „zapytań” nad dokumentami XML
 - wygodny wybór określonych węzłów dokumentu
 - intuicyjna składnia („ścieżki” jak w drzewie katalogów)
 - arytmetyka, porównania
 - bogaty zestaw funkcji
- XPath stosowany w ramach innych standardów
 - W transformacjach XSLT
 - W odsyłaczach XLink
 - W wyrażeniach XPather
 - W schematach XML Schema
 - W pytaniach XQuery

XPath – status

- XPath 1.0 (rekomendacja, listopad 1999)
 - używany m.in. w XSLT 1.0, XML Schema, XPather
- XPath 2.0 (kilka rekomendacji, styczeń 2007)
 - XML Path Language (XPath) 2.0
 - XQuery 1.0 and XPath 2.0 Data Model
 - XQuery 1.0 and XPath 2.0 Functions and Operators
 - XQuery 1.0 and XPath 2.0 Formal Semantics
 - używany w XSLT 2.0
 - złączony z XQuery 1.0
- XPath 3.0
 - Kilka rekomendacji W3C, IV 2014

Dokument XML w modelu XPath

- Dokument jako drzewo
- Uwzględnianie przestrzeni nazw
- Możliwość uwzględniania schematu (w wersji XPath 2.0)
- Rozwinięte sekcje CDATA oraz referencje do encji i znaków
- Brak sąsiadujących węzłów tekstowych
- Atrybut nie jest dzieckiem elementu
- Korzeń, zwany także „węzłem dokumentu”, jest osobnym węzłem, różnym od elementu głównego

XPath 1.0

- Typy danych
 - boolean
 - string
 - number
 - node set
 - w XSLT 1.0 dodatkowa: result tree fragment
- Brak zbiorów wartości prostych
- Zbiory (a nie sekwencje) węzłów

Ograniczenia XPath 1.0

- Prostszy model danych
 - 4 (w XSLT 3) typy danych zamiast typów XML Schema
 - zbiory węzłów zamiast sekwencji wszystkiego
- Brak wyrażenia `if`
- Brak wyrażeń `for`, `some`, `every`
- Brak porównań atomowych
- Mniej testów węzłów
- Mniej standardowych funkcji

Rodzaje węzłów w XPath

- Rodzaje węzłów
 - węzeł dokumentu (korzeń)
 - element
 - atrybut
 - węzeł tekstowy
 - instrukcja przetwarzania
 - komentarz
 - węzeł przesłonięcia nazw
- Brak m.in.
 - sekcji CDATA
 - encji i referencji do encji

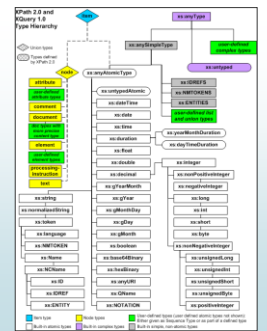
Sekwencje w XPath 2.0

- Elementy (ang. *items*) sekwencji
 - węzły (z drzew dokumentów)
 - wartości atomowe
- Równoważność elementu i jednoelementowej sekwencji
 - $5.55 = (5.55)$
- Spłaszczanie zagnieżdżonych sekwencji
 - $((5.55, (1, 2, 3), 'Aldona') = (5.55, 1, 2, 3, 'Aldona'))$

System typów

- Typy pierwotne XMLSchema
- Dodatkowo
 - xs:untyped
 - xs:untypedAtomic
 - xs:anyAtomicType
 - xs:dayTimeDuration
 - xs:yearMonthDuration
- Możliwość używania zdefiniowanych w schemacie (prosty i złożony), a także innych typów wyprowadzonych
 - o ile przetwarzanie schema aware
 - nie zawsze dostępne

Źródło: <http://www.skachenko.com/blog/archives/000283.html>



Effective Boolean Value

- Często potrzeba traktowania dowolnej wartości jako wartości logicznej
 - `if (cena[waluta]) then`
- Zasady zamiany
 - pusta sekwencja → fałsz
 - sekwencja z węzłem na pierwszej pozycji → prawda
 - pojedyncza wartość boolowska → ta sama wartość
 - pojedynczy pusty napis → fałsz
 - pojedynczy niepusty napis → prawda
 - pojedyncza liczba różna 0 lub NaN → fałsz
 - inna pojedyncza liczba → prawda
 - inna wartość → błąd typu

Literaty i zmienne

- Literaty
 - napisy: '12.5', "He said, ""I don't like it."""
 - liczby: 12, 12.5, 1.13e-8
- Zmienne
 - \$x - referencja do zmiennej o nazwie x
 - zmienne wprowadzane konstrukcjami
 - XPath 2.0 (for, some, every)
 - XSL 1.0/2.0 (variable, param)

13

Operatory

- 68 operatorów XPath (mniej symboli, ale przeciężona notacja)
- Arytmetyka
 - `+`, `-`, `*`, `div`, `idiv`, `mod`
 - na datach i `duration`: `+`, `-` – zgodnie z typami
- Sekwencje węzłów (w 1.0 „zbiory węzłów”)
 - `union()`, `intersect`, `except`
 - nie-węzływ sekwencjach – błąd typu
 - wynik: sekwencja bez powtórzeń, porządek dokumentu
- Wartości logiczne
 - operatorzy `and`, `or`
 - `true()`, `false()`, `not()` to funkcje

14

Operatory porównania

- Porównania atomowe (tylko w XPath 2.0)
 - `eq`, `ne`, `lt`, `le`, `gt`, `ge`
 - na wstępie argumenty poddane atomizacji
 - jeden z argumentów jest sekwencją pustą → wynik jest sekwencją pustą
 - jeden z argumentów jest sekwencją wieloelementową → błąd typów
 - typy atomowe – infuzyjne „normalne” porównanie
 - a le typy pozwalają
- Porównania ogólne (XPath 1.0 i 2.0)
 - `=`, `!=`, `<`, `<=`, `>`, `>=`
 - stosowane do sekwencji
 - istnieje para elementów z lewej i prawej sekwencji, dla której zachodzi odpowiednie porównanie atomowe
 - dla sekwencji jednoelementowych zgodnie z atomowymi

15

Funkcje

- Wywołania funkcji
 - `fn:concat('Pani ', imie, ' ', nazwisko)`
 - `count(//obiekt/@parzysty)`
- 150 standardowych funkcji XPath 2.0, 27 w XPath 1.0
 - w przestrzeni nazw <http://www.w3.org/2005/xpath-functions>
 - domyślna przestrzeń nazw dla funkcji

16

Wybrane funkcje XPath dla tekstu

<code>concat(String ciąg1, String ciąg2, ...)</code>	Składanie łańcuchów tekstów w jedną
<code>contains(String łańcuch, String wzorec)</code>	Sprawdzenie, czy łańcuch zawiera badany wzorec
<code>ends-with(String łańcuch, String wzorec)</code>	Powrót do wylotu podwójny spacji w łańcuchu
<code>starts-with(String łańcuch, String wzorec)</code>	Sprawdzenie, czy łańcuch zaczyna się od określonego wzorca
<code>string-length(String łańcuch)</code>	Długość łańcucha
<code>substring(String łańcuch, Number początek, Number długość)</code>	Wycina określony ciąg z łańcucha znaków
<code>substring-after(String łańcuch, String wzorec)</code>	Zwraca ciąg znaków występujący po określonym wzorcu
<code>substring-before(String łańcuch, String wzorec)</code>	Zwraca ciąg znaków występujący przed określonym wzorcu
<code>substring-index(String łańcuch, String starszaki, String młodszy)</code>	Określa podciąg znaków w określonym łańcuchu tekstów
<code>string()</code>	Umożliwiająca dokonanie konwersji dowolnego obiektu na typ znakowy - String

17

Wybrane funkcje XPath dla liczb

<code>ceiling(Number liczba)</code>	Zwraca liczbę całkowitą, która jest większa bądź równa badanej liczbie
<code>floor(Number liczba)</code>	Zwraca liczbę całkowitą, jednakże mniejszą bądź równą liczbie badanej
<code>round(Number liczba)</code>	Zwraca najbliższą dla badanej liczby liczbę całkowitą
<code>sum(NodeSet zbior)</code>	Zwraca wartość sumowania
<code>number()</code>	Konwersja dowolnego obiektu na typ liczbowy Number

18

Wybrane funkcje XPath dla wartości logicznych

<code>false()</code>	Zwraca wartość false
<code>true()</code>	Zwraca wartość true
<code>not()</code>	Tworzy negację dla wyrażenia logicznego
<code>boolean()</code>	Konwersja dowolnego obiektu na typ logiczny (boolean), zawierający tylko wartości: prawda lub fałsz

19

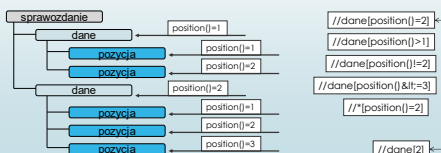
Wybrane funkcje XPath

count(NodeSet element)	Zwraca ilość wystąpień elementu
last()	Zwraca pozycję ostatniego elementu
position()	Zwraca pozycję elementu
id(String identyfikator)	Zwraca węzły mające przypisany identyfikator typu ID
lang(String język)	Porównuje atrybut język z wartością elementu xml:lang
local-name(NodeSet element)	Zwraca lokalną nazwę elementu
name(NodeSet element)	Zwraca nazwę elementu
namespace-uri(NodeSet element)	Zwraca wskazanie na przestrzeń nazw

20

Odwołanie do konkretnego węzła

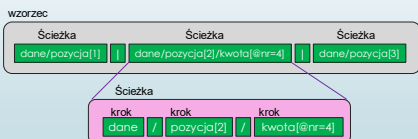
- position()
- Zlicza wyłącznie węzły-dzieci – nie pozwala na sprawdzenie numeru wystąpienia węzła w zagnieżdżonej strukturze



21

Budowa wyrażeń XPath

- Najmniejsza jednostka wchodząca w skład języka - krok
- Złożenie wielu kroków składa się na ścieżkę – poprzedzielane symbolem „/”
- Utworzenie alternatywnych ścieżek powoduje utworzenie wzorca



22

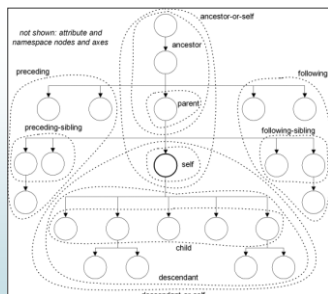
Location path

- Ścieżka
 - Bezwzględna /krok/krok ...
 - Względna krok/krok ...
- Krok – składnia w pełni rozwinięta
 - oś::test-węzłów [predykat1] [predykat2] ...
 - oś – kierunek w drzewie dokumentu
 - test-węzłów – wybór węzłów po rodzaju, typie, nazwie
 - predykat – opcjonalny, dodatkowo filtrujący węzły

```
/child::book/child::section
child::para[attribute::type="waluta"]
```

23

Osie



Źródło: <http://www.georgehnanides.com/html/Computers/XML/XSL/XPath.asp>

24

Osie

Słowo języka XPath	Znaczenie
child	Węzły-dzieci
descendant	Węzły-potomkowe
parent	Węzeł-rodzic (pusty dla węzła korzenia)
ancestor	Węzły-przodkowie
following-sibling	Kolejne węzły na tym samym poziomie
preceding-sibling	Poprzednie węzły na tym samym poziomie
following	Wszystkie kolejne węzły (bez potomków)
preceding	Wszystkie poprzednie węzły (bez potomków)
self	Aktualny węzeł
descendant-or-self	Potomkowie oraz aktualny węzeł
ancestor-or-self	Przodkowie oraz aktualny węzeł
attribute	Atrybut znajdujący się w badanym węźle
namespace	Przebież nazw

25

Przykłady

- `self::*` – dostęp do aktualnego węzła
- `preceding::pozycja` – umożliwia dostęp do poprzedzających aktualny węzeł elementów o nazwie `pozycja`
- `preceding::text()` – umożliwia dostęp do poprzedzających aktualny węzeł zawartości tekstowych
- `following-sibling::node()` – umożliwia dostęp do elementów występujących na tym samym poziomie, co aktualny węzeł i będących jego następnikami
- `following-sibling::comment()` – umożliwia dostęp do komentarzy występujących na tym samym poziomie, co aktualny węzeł i będących jego następnikami
- `ancestor::processing-instruction()` – umożliwia dostęp do instrukcji przetwarzania występujących w węzłach-przodkach
- `parent::node()/attribute::status` – umożliwia dostęp do atrybutu `status` znajdującego się w węźle-rodzicu

26

Skróty składniowe

Oś	Skrótowy odpowiednik	Znaczenie
<code>child::</code>		Dzieci aktualnego węzła
<code>parent::node()</code>	<code>..</code>	Rodzic aktualnego węzła
<code>descendant-or-self::node()</code>	<code>..</code>	Potomkowie lub aktualny węzeł
<code>self::node()</code>	<code>.</code>	Aktualny węzeł
<code>attribute::</code>	<code>@</code>	Atrybut

```

./para
self::node()/descendant-or-self::node()/child::para

```

27

Dodatkowe testy węzłów w XPath 2.0

- `document-node()`
- `element()`
- `element(osoba)`
- `element(*, osobaTyp)`
- `element(osoba, osobaTyp)`
- `attribute()`
- `attribute(id)`
- `attribute(*, xs:integer)`
- `attribute(id, xs:integer)`

28

Predykaty

- Utworzenie ścieżki, która wystarczająco zawęża położenie węzłów
- Zbudowanie wyrażenia zawierającego kryteria warunkowe, tzw. wyrażenie warunkowe
- Obliczane dla każdego węzła (węzeł na chwilę staje się węzłem kontekstowym)
- Każdy predykat „przesiewa” sekwencję
- W zależności od typu wyniku predykatu
 - liczba – porównywana z pozycją węzła w sekwencji (od 1)
 - nie liczba – decyduje Effective Boolean Value
- Możliwe użycie poza ścieżkami (tzw. *filter expressions*)

```

/child::osoby/child::osoba[child::imię = 'Patrik']
child::osoba[child::imię = 'Patrik']/child::nazwisko
//obiekt[attribute::nazwa][3]

```

29

Sposób obliczania ścieżek

- Ścieżki obliczane są od lewej do prawej
- Dla każdego węzła z bieżącej sekwencji obliczany kolejny krok (wraz z predykatami)
 - `//oddział/pracownik[1]`
- Po każdym predykanie zbierana jest cała sekwencja, przekazywana do kolejnego predykatu (zmiana kontekstu)
 - `//pracownik[@kierownik and position() = 5]`
 - `//pracownik[@kierownik][5]`

30

Przykłady

```

//dane/pozycja[okres='Styczeń']
//dane/pozycja[@waluta != 'PLN']
//okres[.='Luty']
//dane/pozycja/wartość[. < 1000]
//pozycja[wartość mod 1000=0]
//pozycja[wartość*1.22 > 1000]
//pozycja[1]/wartość + //pozycja[2]/wartość
following-sibling::*[1]/wartość div preceding-sibling::*[1]/wartość
//pozycja[contains(okres, 'erwiec')]
//pozycja[@*='miesiac']/okres

```

31

Wyrażenia warunkowe (XPath 2.0)

```
if WARUNEK
  then WYNIK1
  else WYNIK2
```

- Liczy się Effective Boolean Value
- Obliczana tylko jedna gałąź

```
if informacje/cena
then
  if informacje/cena >= 1000
  then 'Przesyłka wartościowa'
  else 'Przesyłka zwykła'
else 'Brak danych'
```

32

Pętla po sekwencji (XPath 2.0)

```
for $ZMIENNA in SEKWENCJA
  return WYNIK
```

- ZMIENNEJ przypisywane są kolejne wartości z SEKWENCJI
- WYNIK obliczany jest z wybraną wartością ZMIENNEJ
- wynik całości – sekwencja wyników częściowych

```
for $i in (1 to 10)
  return $i * $i

for $o in //obiekt
  return concat('Obiekt:', $o/@nazwa)
```

33

Kwantyfikatory (XPath 2.0)

```
some $ZMIENNA in SEKWENCJA
  satisfies WARUNEK
every $ZMIENNA in SEKWENCJA
  satisfies WARUNEK
```

- Liczy się Effective Boolean Value
- Dopuszczalna kolejność przechodzenia po sekwencji

```
some $i in (1 to 10) satisfies $i > 7
every $o in //obiekt satisfies $o/@nazwa
```