Przetwarzanie dokumentów XML i zaawansowane techniki WWW

"Wprowadzenie do walidacji dokumentów XML przez XML-Schema"

(Zajęcia 04 - 21.03.2016 r.)

1) Walidacja przez schematy - XML Schema

Poznaliśmy już walidację dokumentów XML w modelu DTD (Document Type Definition), gdzie definiowaliśmy zestaw reguł opisujących występowanie i zachowanie elementów oraz ich atrybutów. W tym modelu możliwe było sformułowanie jaka struktura jest prawidłowa dla dokumentu XML. Niestety zastosowanie DTD miało znaczące ograniczenia wynikające ubogiego zestawu reguł jakie można było tworzyć dla danego pliku XML.

Modelem walidacji dokumentów XML z bogatym zestawem reguł jest "XML Schema" - smechaty XML. Pozwalają one również na tworzenie definicji typów dokumentów, czyli struktur opisujących wygląd dokumentu XML jednak z dużo większymi możliwościami.

Schematy XML są technologia walidacji rekomentowaną i zalecaną przez organizację standaryzująca W3C i jej opis można znaleźć własnie na ich stronie: www.w3schools.com/schema oraz link do referencji z opisem opcji reguł na elementach:
http://www.w3schools.com/schema/schema_elements_ref.asp

Podstawową cechą dokumentu ze schematem jest to że sam on sam jest dokumentem XML!!!
Jest oparty na przestrzeni nazw XS oraz XSD:

Przestrzenie te są zdefiniowane w pod adresami: http://www.w3.org/2001/XMLSchema
Schematy walidujące są przechowywane w osobnych plikach zwykle z rozszerzeniem "plik.xsd".

Pierwszy schemat oraz ogólna struktura:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <!--  Definicje elementów i atrybutów / definicje typów nazwanych i
nienazwanych. -->
</xs:schema>
```

Odwołanie się do schematu w pliku XML w elemencie głównym:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="plik.xsd">
</root>
```

Metoda walidacji plików XML razem ze Schematem XML:

1) z konsoli w Linux:

```
> xmllint --schema   file.xsd   plik.xml
```

2) walidator on-line:

http://www.utilities-online.info/xsdvalidation/

Przy tworzeniu schematów XML wykorzystujemy tzw. “TYPY” opisujące złożone węzły występujące w naszych XML-ach. Typy te są swoista strukturą opisująca zawartość poszczególnych elementów w pliku XML. Typy dzielimy na dwie grupy:

a) nazwane: używamy wtedy kiedy przewidujemy, że kilka elementów może być tego samego typu oraz posiada ta samą strukturę zawartości.

b) nienazwane: zwykle kiedy tylko jeden element jest opisany danym typem.

W pliku będącym schematem podobnie jak w DTD musimy zadbać o opis każdego z elementów jakie znajdują się w naszym pliku XML / lub mogą tam wystąpić.

Należy również rozróżnić elementy na złożone oraz proste:

a) elementy złożone: to takie które zawierają inne lementy potomne,

b) elementy proste: to takie które nie zawierają innych elementów a jedynie dane różnych typów.

Deklaracje elementów:

```
<xs:element name=”nazwa_elementu”>
     <!-- tu opisujemy dany element w zależności jaki to jest element →
</xs:element>
```

lub w przypadku posiadania typu nazwanego:

```
<xs:element name=”nazwa_elementu” type=”nazwa_typu” />
```

Gdy element jest typu prostego i nie zawiera już w sobie innych elementów a jedynie dane opisujemy go w nastepujący sposób:

```
<xs:element name=”nazwa_elementu” type=”xs:type”/>

xs:type = xs:string, xs:integer, xs:decimal, xs:boolean, xs:date, xs:time.
```

Dodatkowo przy elementach prostych można zdefiniować wartośc domyślną lub ustalona elementu:

```
<xs:element name="nazwa_elementu" type="xs:type" default="wartosc_domyslna"/>
<xs:element name="nazwa_elementu" type=”xs:type"
fixed="wartosc_stala-nadana"/>
```

Definicja atrybutów:

Podobnie jak w przypadku elementów atrybuty moga być typu prostego lub “prostego z restrykcjami”, i w takim przypadku można dla nich zdefiniowac osoby “typ prosty”.

Definicja atryputów jest realizowana przez dyrektywę:

```
<xs:attribute name=”nazwa_atrybutu” type=”xs:type”>
```

lub

```
<xs:attribute name=”nazwa_atrybutu” type=”nazwa_typu”>
```

Podobnie jak w przypadku elementów atrybuty mogą posiadać wartości domyślną oraz ustaloną:

```
<xs:attribute name="nazwa_elementu" type="xs:type"
default="wartosc_domyslna"/>
<xs:attribute name="nazwa_elementu" type="xs:type"
fixed="wartosc_stala-nadana"/>
```

Atrybuty z definicji są opcjonalne w elementach dokumentów XML, jednak kiedy istnieje potrzeba aby atrybut pojawił się w definicji elementu możemy to wymusić przez regułę:

```
<xs:attribute name="nazwa_elementu" type="xs:type" use="required"/>
```

Czasem dla typów prastych pojawia się potrzeba zdefiniowania dodatkowych warunków lub ograniczenia występujących wartości. Korzysta się wtedy z definicji “typu prostego” który znów może być nazwany lub nienazwany. Typ prosty nienazwany definiujemy w następujący sposob:

```
<xs:simpleType>
      definicja
</xs:simpleType>
```

natomiast typ nazwany :

```
<xs:simpleType name=”nazwa_typu”>
      definicja
</xs:simpleType>
```

Natomiast typy założone definiowane są w podobny sposób:

```
<xs:complexType>
      definicja
</xs:complexType>
```

natomiast typ nazwany :

```
<xs:complexType name=”nazwa_typu”>
      definicja
</xs:complexType>
```

Dodatkowo warto wsponieć że najczęściej w definicji typów złożonych pojawiją się definicje:

1) sequence: o kolejności występowania elementów w dokumencie XML:

```
<xs:sequence>
    <xs:element name="element_1" type="xs:string"/>
    <xs:element name="element_2" type="xs:string"/>
 </xs:sequence>
```

2) all: musza wystąpić raz w dowolnej koljności

```
<xs:all>
      <xs:element name="element_1" type="xs:string"/>
      <xs:element name="element_1" type="xs:string"/>
 </xs:all>
```

3) choice: może wystąpić albo jeden element albo drugi:

```
<xs:choice>
      <xs:element name="element_1" type="type1"/>
      <xs:element name="element_1" type="type2"/>
</xs:choice>
```

Występowanie danego elementu:

1) maksymalna liczba wystąpień elementu danego typu:

```
      <xs:element name="child_name" type="xs:string" maxOccurs="10"/>
```

2) maksymalna i minimalna liczba wystąpień elementów danego typu

```
      <xs:element name="child_name" type="xs:string" maxOccurs="10"
minOccurs="0"/>
```

3) przewidzenie wystąpienia elementu nie opisnaego schematem:

```
<xs:any minOccurs="0"/>
```

4) przewidzenie wystąpienia atrybutu elementu nie przewidzianego schematem:

```
<xs:anyAttribute/>
```

Restrykcje wartości oraz zakresu występowalności elementów i atrybutów:

```
<xs:restriction base="xs:typ_na_jaki_kladziemy_ograniczenie">
        definicja
</xs:restriction>
```

najważniejsze możliwości to:

1) ograniczenie wartości liczbowej:

```
<xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="20"/>
</xs:restriction>
```

2) Ograniczenie zakresu możliwych wartości elementu (typ wyliczeniowy):

```
<xs:restriction base="xs:string">
      <xs:enumeration value="zielony"/>
      <xs:enumeration value="czerwony"/>
      <xs:enumeration value="niebieski"/>
 </xs:restriction>
```

3) ograniczenie przez podanie wzorca wyrażenia regularnego:

```
<xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
</xs:restriction>
```

akceptujemy pieć liczb po od 0-9

```
<xs:restriction base="xs:integer">
      <xs:pattern value="[0-9][0-9][0-9][0-9][0-9]"/>
</xs:restriction>
```

4) ogrniczenie długości wpisywanych wartości:

```
<xs:restriction base="xs:string">
      <xs:length value="8"/>
</xs:restriction>
```

lub

```
<xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
</xs:restriction>
```

Przykład:
Weźmy plik XML z prosta walidacją DTD i przeróbmy go na walidację przez schemat:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ksiazka [
<!ELEMENT ksiazka (autor,tytul,datawydania,wydawca)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT tytyl (#PCDATA)>
```

```
<!ELEMENT datawydania (#PCDATA)>
<!ELEMENT wydawca (#PCDATA)>
]>
<ksiazka>
     <autor>Adam Mickiewicz</autor>
     <tytul>Pan Tadeusz</tytul>
     <datawydania>1864-11-01</datawydania>
     <wydawca>Wydawnictwo Chochol</wydawca>
</ksiazka>
```

Tworzymy schemat w domenie typów nienazwanych:

```
 <?xml version="1.0" encoding="UTF-8"?>
   <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="ksiazka">
       <xs:complexType>
          <xs:sequence>
              <xs:element name="autor" type="xs:string"/>
              <xs:element name="tytul" type="xs:string"/>
              <xs:element name="datawydania" type="xs:string"/>
              <xs:element name="wydawca" type="xs:string"/>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
</xs:schema>
```

Przykład schematu zdefiniowanego przez nazwane typy danych:

Plik XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<osoby xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="osoba.xsd">
     <osoba plec="m" email="jan@kowalski.pl">
        <imie>Jan</imie>
        <nazwisko>Kowalski</nazwisko>
     </osoba>
     <osoba plec="k" email="anna@kowalski.pl">
        <imie>Anna</imie>
        <nazwisko>Kowalska</nazwisko>
     </osoba>
     <osoba plec="w">
        <imie>Robert</imie>
        <nazwisko>Kowalski</nazwisko>
     </osoba>
</osoby>
```

Plik schematu:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!-- anaoacja opisujaca schemat -->
    <xs:annotation>
          <xs:documentation xml:lang="pl">
```

```
            Przykład schematu z definicjami na poziomie głównym i typami
nazwanymi.
          </xs:documentation>
    </xs:annotation>

    <!-- Definicje elementów i atrybutów na poziomie głównym. -->
    <xs:element name="osoby" type="OsobyTyp"/>
    <xs:element name="osoba" type="OsobaTyp"/>
    <xs:element name="imie" type="xs:string"/>
    <xs:element name="nazwisko" type="xs:string"/>
    <xs:attribute name="plec" type="PlecTyp"/>
    <xs:attribute name="email" type="xs:string"/>

    <!-- Definicje typów na poziomie głównym. -->
    <xs:complexType name="OsobyTyp">
       <xs:sequence>
         <xs:element ref="osoba" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
    </xs:complexType>

    <xs:complexType name="OsobaTyp">
       <xs:sequence>
         <xs:element ref="imie" minOccurs="1" maxOccurs="unbounded"/>
         <xs:element ref="nazwisko"/>
       </xs:sequence>
       <xs:attribute ref="plec" use="required"/>
       <xs:attribute ref="email" use="optional"/>
    </xs:complexType>

    <xs:simpleType name="PlecTyp">
       <xs:restriction base="xs:string">
         <xs:enumeration value="k"/>
         <xs:enumeration value="m"/>
       </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

Zadanie 1:

Proszę dla przygotowanego na poprzednich zajęciach pliku XML utworzyć plik Schemat definiujący strukturę.