Hipertekst i Hipermedia
Projekt (2013)

Celem projektu jest zapoznanie studentów z hipertekstem i hipermediami. Szczególny nacisk położono na prezentację informacji w środowisku internetowym (WWW) z wykorzystaniem języka HTML oraz XML.

Temat:

MOJA PASJA

Etapy:

Etap	Punktacja [pkt]
HTML, Dokument XML, XML Schema	50
XSLT + FO	50

Etap 1: HTML (20pkt), Dokument XML, XML Schema (30pkt)

HTML: (20 pkt)
Wymagania:
- zawartość strony zgodna z tematem projektu
- walidowanie strony (**2pkt**)
- układ strony:
  - podział strony na kilka elementów (nagłówek, menu, stopka, pole z treścią) (**3pkt**)
- rozdzielenie treści na kilka plików (przynajmniej trzy) (**2pkt**)
- menu zawierające przynajmniej trzy opcje a jedna z nich z dodatkowymi opcjami podrzędnymi; zaznaczanie wybranej opcji (**1pkt**)
- umieszczenie na stronie multimediów:
  - grafika
    - galeria zdjęć (przynajmniej 5) ma być zorganizowana w postaci miniaturek, które można obejrzeć powiększone (**2pkt**)
- umieszczenie na stronie:
  - tabeli (**1pkt**)
  - adresu e-mail z możliwością wysłania poczty (**1pkt**)
  - odsyłaczy do innych stron internetowych (**1pkt**)
  - odsyłacza do wybranego miejsca w tekście lub do początku strony (wyświetlony tekst powinien być odpowiednio długi, aby była możliwość zademonstrowania tej opcji) (**1pkt**)
- style należy zdefiniować w oddzielnym arkuszu stylów, wykorzystać mechanizm CSS
  - różne style dla przynajmniej 4 selektorów (grup selektorów) (**2pkt**)
  - identyfikatory lub klasy (przynajmniej 4) (**2pkt**)
- stworzenie prostej ankiety-formularza (**2pkt**)
  - przynajmniej 3 różne rodzaje pól umożliwiających wprowadzanie danych,
  - przyciski do czyszczenia zawartości formularza oraz wysyłania danych
- dbałość o estetyczny wygląd strony

XML, XML SCHEMA: (30pkt)
Wymagania:
- Utworzyć plik w formacie XML zawierający dane związane z tematem projektu. W pliku muszą znaleźć się zdjęcia oraz linki.
- Dla pliku XML, aby wymusić jego odpowiednią składnię, należy zaprojektować i utworzyć plik XML Schema.

- Plik XML musi być poprawny składniowo i semantycznie. Struktura pliku XML musi być zgodna z podaną w XML Schema. Do sprawdzenia poprawności należy użyć walidatora (**http://tools.decisionsoft.com/schemaValidate/**).
- Dla stworzonego pliku XML wygenerować XML Schema przy użyciu Visual C++. Na zaliczenie projektu należy przynieść zarówno XML Schema stworzony przez siebie, jak i wygenerowany automatycznie.
- Należy również zwrócić uwagę na postać dokumentu, czyli sposób zapisu, stosowanie wcięć obrazujących strukturę danych, odpowiednie (adekwatne do zawartej w nich treści) nazywanie znaczników, atrybutów.

Wymagania szczegółowe:

W pliku XML Schema należy zadeklarować i wykorzystać:

- co najmniej 6 definicji globalnych typów złożonych (**4pkt**)
- przynajmniej 5 definicji globalnych typów prostych (**4pkt**)
- co najmniej 2 definicje lokalnych typów złożonych (**2pkt**)
- przynajmniej 2 definicje lokalnych typów prostych (**2pkt**)
- przynajmniej jedna definicja grupy (elementów lub atrybutów) (**1pkt**)
- istnienie przynajmniej 4 poziomów zagłębienia w strukturze dokumentu (**2pkt**)
- definicja przynajmniej 5 atrybutów z czego przynajmniej 1 zdefiniowany globalnie i użyty przynajmniej 2 razy (**3pkt**)
- definicja przynajmniej 10 różnych elementów (**4pkt**)
- stosowanie aspektów (ograniczeń na elementy i atrybuty) (**3pkt**)
  - *length, minLength, maxLength, maxInclusive, minInclusive,maxExclusive, minExclusive,* (wybrane min 3)
  - *pattern, enumeration*
- wyprowadzanie typów (**1pkt**)
  - *extension*
- przynajmniej 3 odnośniki do elementów i/lub atrybutów (ma być odniesienie i do atrybutu i do elementu) (**2pkt**)
- użycie listy (**1pkt**)
- wykorzystanie kombinacji (*union*) (**1pkt**)
- walidowanie pliku

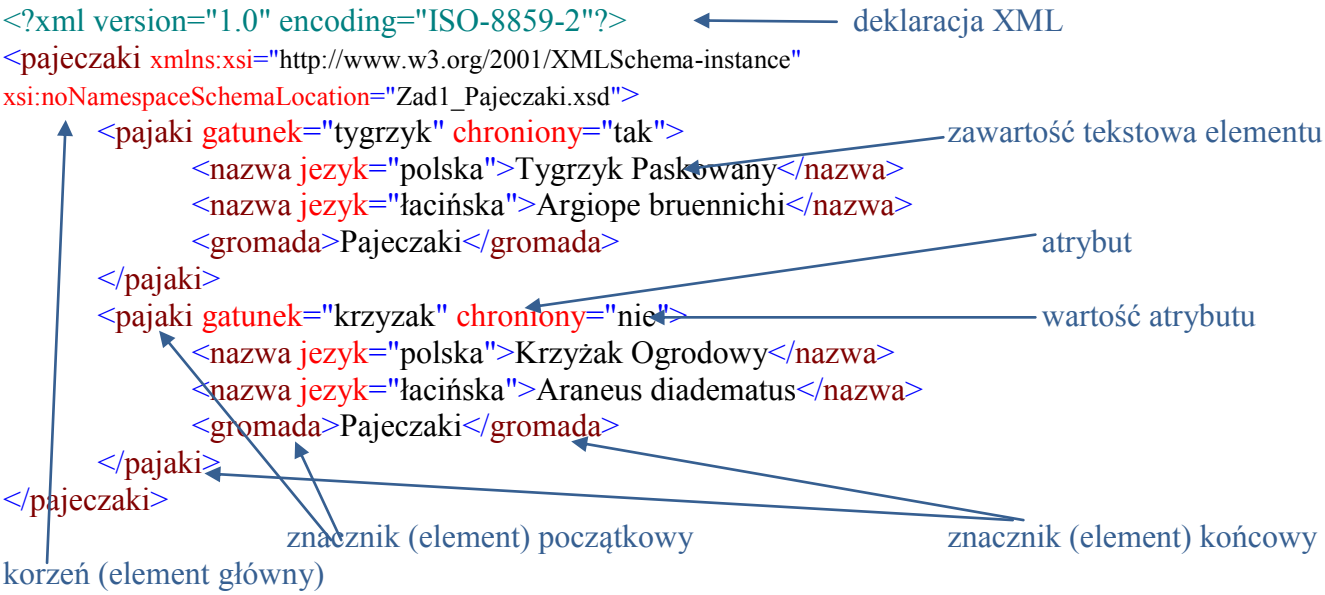
Wybrane przykładowe błędy występujące w schematach:

- błędy walidacji (plik się nie waliduje) (**do -25pkt**)
- trywialna definicja typu prostego (np. typ prosty, który jest zwykłym typem string) (**-4pkt**)
- trywialna definicja typu złożonego (**-4pkt**)
- powtarzanie definicji typów (**-4pkt**) (wielokrotne definiowanie typów)
- bezsensowne nazwy elementów i atrybutów (**-3pkt**)
- wykorzystanie anyType (**do -20pkt**)
- nieznacznie przerobiony, wygenerowany plik xsd (**do -15pkt**)
- niepoprawne definiowanie elementów, atrybutów, struktury (**do -10pkt**)
  - np. zamiast używać maxOccurs=4, czterokrotne definiowanie takiego samego elementu
- brak zdjęć (w XML (min 4) oraz w Schema) (**-1pkt**)
- brak linków (w XML (min 4) oraz w Schema) (**-1pkt**)

XML i XML Schema - krótka ściąga ☺

XML

- wszystkie niepuste elementy muszą mieć znacznik początkowy i końcowy
- elementy mogą być zagnieżdżone, nie mogą na siebie zachodzić
- rozróżnianie dużych i małych liter



XML Schema

Jeśli chcemy stworzyć:

- tylko element z zawartością tekstową
  - typ prosty
- element z podelementami
  - typ złożony
- element z podelementami i atrybutami
  - typ złożony
- element z zawartością mieszaną (podelementy i tekst)
  - typ złożony z atrybutem mixed=true
- element z atrybutami
  - typ złożony
- element z atrybutami i zawartością tekstową
  - simpleContent

1) Definicja typu prostego nazwanego

```
<xs:simpleType name=”krotki_string”>          typ prosty nazwany
    < xs:restriction base=“string”/>          ograniczenie
        < xs:maxLength value=“20” />
    </ xs:restriction>
</ xs:simpleType>
```

2) Definicja elementu

```
<xs:element name="pajaki" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="nazwa" maxOccurs="unbounded">
            <xs:complexType >
               <xs:attribute name="jezyk" type="xs:string" />
            </xs:complexType>
         </xs:element>
         <xs:element name="gromada" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="gatunek" type="xs:string" />
      <xs:attribute name="chroniony" type="xs:string" />
   </xs:complexType>
</xs:element>
```

liczba wystąpień

definicja elementu

typ złożony, lokalny

sekwencja, elementy w ściśle określonej kolejności

typ atrybutu

definicja atrybutu (zawsze po definicjach elementów)

3) Wyliczenia - lista predefiniowanych wartości

```
<xs:simpleType name="nazwa_typu" >
  < xs:restriction base="string">
     < xs:enumeration value="wartosc1" />
     < xs:enumeration value=" wartosc2" />
     < xs:enumeration value=" wartosc3" />
  </ xs:restriction>
</ xs:simpleType>
```

4) SimpleContent

Gdy tworzymy pochodny typ złożony na podstawie typu prostego lub innego typu złożonego o zawartości prostej. Można w ten sposób np. dodać atrybuty do typu bazowego.

```
<xs:complexType name="nazwa_typu">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="nazwa_atrybutu" type="xs:string"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

5) Odniesienia do elementu

```
<xs:element name="data" type="xs:date"/>
```

globalna definicja elementu

```
<xs:element ref="data" minOccurs="0"/>
```

odniesienie do elementu zdefiniowanego globalnie