

Semestrální práce ke kurzu 4iz238 - Dodací list

Oldřich Koutecký

Semestrální práce ke kurzu 4iz238 - Dodací list

Oldřich Koutecký

Table of Contents

Úvod	iv
1. XML dokument, schema a schematron	1
XML dokument	1
XML schema	1
Schematron	2
2. Transformace	3
HTML transformace	3
PDF transformace	4
3. Dokumentace	6
Index	7

Úvod

V této dokumentaci jsou popsány všechny XML soubory vytvořené pro zpracování semestrální práce ke kurzu 4IZ238. Předmětem bylo vytvoření XML dokumentu, schématu, schematronu, transformací do PDF a HTML dokumentu a nakonec samotná dokumentace v DOCBOOKU.

Chapter 1. XML dokument, schema a schematron

XML dokument

Pro téma mé semestrální práce jsem se rozhodl pro zpracování dodacích listů objednávky pro prodejce horolezeckého vybavení. V XML dokumentu sleduji informace o dodavateli, fakturačním a dodacím zákazníkovi a nakonec o zboží, které si obchod objednal.

XML schema

Testovací dokument katalog.xml je validován pomocí XML schematu schema.xsd toto schema bylo navrženo ve struktuře slepého benáťčana, který se mi zdál jako nejlepší volba pro rychlé aktualizace případných oprav či úprav.

V mé práci se převážně objevují typy kontrol pomocí enumerationu či patternu následuje ukázka kódu těchto omezení :

V tomto omezení nastavuji, že hodnoty vložené do elementu Pohlaví mohou být pouze ty které jsou zde uvedeny.

```
<xs:simpleType name="PohlavíType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="pánské"/>
    <xs:enumeration value="dámské"/>
    <xs:enumeration value="univerzální"/>
  </xs:restriction>
</xs:simpleType>
```

A naopak v tomto omezení nastavuji že hodnoty vkládané do elementu identifikační číslo musí být ve stylu - 10 číslic a 5 velkých písmen neboli 0203222101NEKSP

```
<xs:simpleType name="IdentifikačníČíslo">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{10}\D*\p{Lu}{5}"></xs:pattern>
  </xs:restriction>
</xs:simpleType>
```

Pomocí referenční integrity neboli unikátního klíče jsem zajistil že se v dokumentu nebude objevovat stejný typ zákazníka (jedná se o atribut elementu zákazník: dodací nebo fakturační). Ukázka kódu:

```
<xs:complexType name="dodacíListType">
  <xs:sequence>
    <xs:element type="metadataType" name="metadata"/>
    <xs:element type="HeaderType" name="Header">
      <xs:unique name="zakaznikKontrola">
        <xs:selector xpath="mstns:Zakaznik"/>
        <xs:field xpath="@typ"/>
      </xs:unique>
    </xs:element>
    <xs:element type="PoložkyType" name="Položky"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:sequence>  
</xs:complexType>
```

Schematron

Jako další požadavek bylo vytvoření schematronu který bylo možno vytvořit v rámci schematu či jako samostatný soubor. Já jsem se rozhodl pro druhou možnost tudíž mou práci můžete vidět zde `schematron.sch` v tomto schematronu kontroluji například minimální množství položky nutné pro objednání. Zde můžete vidět ukázkou kódu :

```
<sch:pattern>  
  <sch:title>Minimální množství objednávky</sch:title>  
  <sch:rule context="o:Položky">  
    <sch:assert test="if(o:Položka/o:Mnozstvi/@jednotky='kusy') then(1)">  
  </sch:rule>  
</sch:pattern>
```

Chapter 2. Transformace

Dalším úkolem této práce bylo transformování XML dokumentu do HTML a PDF dokumentu. K tomuto transformování jsou používány XSL styly

HTML transformace

Pro transformaci do HTML dokumentu jsem vytvořil soubor `html.xsl` v tomto souboru jsou uvedeny šablony pro formátování a zobrazení XML dokumentu v HTML podobě. V následujícím kódu ukáži vygenerování objednaných položek do tabulky a následné vygenerování samostatné HTML stránky pro každou položku :

```
<xsl:for-each select="//Položka">
  <a href="polozka{position()}.html">
    <xsl:value-of select="IdentifikačníČíslo"/> - <xsl:value-of select="..."
  </a>
  <br/>
</xsl:for-each>
</xsl:template>

<xsl:template match="s:Položky">
<table>
<caption>
  <h2>Seznam položek</h2>
</caption>
<tr>
  <th>Identifikační číslo</th>
  <th>Popis</th>
  <th>Datum výroby</th>
  <th>Datum expirace</th>
  <th>Pečeť</th>
  <th>Šarže</th>
  <th>Jednotková cena</th>
  <th>Množství</th>
  <th>Počet balení</th>
  <th>QR Kód</th>
  <th>TXT</th>
</tr>
<xsl:for-each select="//s:dodacíList/s:Položky/s:Položka">
  <tr>
    <td>
      <xsl:value-of select="s:IdentifikačníČíslo"/>
    </td>
    <td>
      <xsl:value-of select="s:Popis"/>
    </td>
    <td>
      <xsl:value-of select="s:Barva"/>
    </td>
    <td>
      <xsl:value-of select="s:Materiál"/>
    </td>
    <td>
      <xsl:value-of select="s:Pohlaví"/>
    </td>
```

```

        <td>
            <xsl:value-of select="s:Velikost"/>
        </td>
        <td><xsl:value-of select="s:CenaZaKus"/>&#160;<xsl:value-of
            select="s:CenaZaKus/@měna"/></td>
        <td><xsl:value-of select="s:Mnozstvi"/>&#160;<xsl:value-of
            select="s:Mnozstvi/@jednotky"/></td>
        <td>
            <xsl:value-of select="Značka"/>
        </td>
        <td>
            <xsl:choose>
                <xsl:when test="FotoURI[not(text())]">Náhled není k dispo
                <xsl:otherwise>
                    <img alt="Foto položky">
                        <xsl:attribute name="src">
                            <xsl:text>png/</xsl:text>
                            <xsl:value-of select="FotoURI"/>
                        </xsl:attribute>
                    </img>
                </xsl:otherwise>
            </xsl:choose>
        </td>
        <td>
            <xsl:value-of select="s:TXT"/>
        </td>
    </tr>
</xsl:for-each>

```

Pro uživatelsky přívětivější zobrazení HTML stránky jsem vytvořil kaskádový styl `style.css`. Tento dokument pouze upravuje pozicování tabulek a nadpisů. Ukázka kódu :

```

table {
    border-collapse: collapse;
    float: left;
    border: solid;
    margin: 20px;
}

img {
    width: 150px;
    height: 150px;
}

td {
    border: 1px solid #999;
    padding: 0.5rem;
    text-align: left;
}

```

PDF transformace

Jako předposlední část této práce je transformace XML dokumentu do PDF. Soubor kterým tuto transformaci provádím je `pdf.xsl`. Rozdíl mezi HTML transformací a touto je ten že se zde používají formátovatelné bloky `[fo:block]`. Ve výsledném dokumentu jsem se snažil zachovat podobný styl jako v HTML transformaci. Ukázka kódu :


```
<fo:static-content flow-name="xsl-region-before">
  <fo:block text-align="center" margin-top="5mm">
    <xsl:text>Zákazníková objednávka číslo : </xsl:text><xsl:value-of
  </fo:block>
</fo:static-content>

<fo:static-content flow-name="xsl-region-after">
  <fo:block text-align="center">
    <xsl:text>Strana </xsl:text>
    <fo:page-number/>
    <xsl:text> / </xsl:text>
    <fo:page-number-citation ref-id="posledni"/>
  </fo:block>
</fo:static-content>
```

V tomto kódu vytvářím záhlaví a zápatí. V záhlaví je uvedeno číslo objednávky a den kdy bylo toto PDF vtištěno a v zápatí je uvedena informace o počtu stran dokumentu.

Chapter 3. Dokumentace

Jako poslední částí bylo vytvoření dokumentace pomocí DOCBOOKU a následné vygenerování tohoto zpracování do podoby HTML a PDF.

Index

C

CSS, 4

D

DOCBOOK, 6

H

HTML transformace, 3

P

PDF transformace, 4

R

Referenční integrita, 1

S

Schematron, 2

T

Transformace, 3

X

XML, 1

XML schematu, 1