

Semantic Web

XML Schema, CSS

F. Abel, D. Krause

IVS Semantic Web Group

23.10.2008

StudiVZ¹ plans to make data accessible via RESTful API, which should return XML. Define an XML Schema that models the following:

- the result of the StudiVZ API should return a list of students
- every student has a username, date of birth, and mail address
- a student has a name, which consists of given name and surname
- every student has a gender, which is either female or male
- students have friends, which are also students
- there are special students, graduated students, that have a certain degree and a final grade (between 1 and 5)

¹<http://www.studivz.net>

Solution - Exercise 1: XML Schema (1)

XML Schema Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.studivz.net/studiSchema"
  elementFormDefault="qualified"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:stu="http://www.studivz.net/studiSchema">
  <complexType name="person">
    <sequence>
      <element name="name" type="stu:name"/>
      <element name="friend" type="IDREF" maxOccurs="unbounded" minOccurs="0"/>
    </sequence>
    <attribute name="date-of-birth" type="date" use="required"></attribute>
    <attribute name="username" type="ID" use="required"></attribute>
    <attribute name="mail" type="string" use="required"></attribute>
    <attribute name="gender" type="stu:gender" use="optional"></attribute>
  </complexType>
  <complexType name="name">
    <attribute name="firstname" type="string"></attribute>
    <attribute name="lastname" type="string"></attribute>
  </complexType>

  <element name="studenten" type="stu:personen"/>
  ...

```

Solution - Exercise 1: XML Schema (2)

```
...
<complexType name="personen">
  <sequence>
    <element name="student" type="stu:person" minOccurs="1"
maxOccurs="unbounded"></element>
  </sequence>
</complexType>

<simpleType name="gender">
  <restriction base="string">
    <enumeration value="female"></enumeration>
    <enumeration value="male"></enumeration>
  </restriction>
</simpleType>

<complexType name="graduatedPerson">
  <complexContent>
    <extension base="stu:person">
      <attribute name="degree" type="string"></attribute>
      <attribute name="finalGrade" type="stu:grade"></attribute>
    </extension>
  </complexContent>
</complexType>
...
```

...

```
<simpleType name="grade">  
  <restriction base="double">  
    <maxInclusive value="5"></maxInclusive>  
    <minInclusive value="1"></minInclusive>  
  </restriction>  
</simpleType>  
</schema>
```

Exercise 2: XML, XML Schema

Correct the errors in the given XML document below so that it is valid regarding the XML Schema you defined above!

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="studi-style.css"?>
<stu:studenten
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.studivz.net/studiSchema studivz.xsd">
  <stu:student
    gender="female" mail="anna@web.de"
    username="anna" xsi:type="stu:person">
    <stu:name firstname="Anna" lastname="Buttermann">
    <stu:friend id="rudi"/>
    <stu:friend id="rudi"/>
  </stu:student>
  <stu:student date-of-birth="1980-07-04" gender="male" mail="peter@web.de"
    username="rudi">
    <stu:name firstname="Peter" lastname="Brotmann"/>
    <stu:friend>anna</stu:friend>
  </stu:student>
  <stu:student finalGrade="0.7" degree="Dipl. Ing."
    date-of-birth="1950-07-08" username="rudi" gender="mann"
    name="Rudi Wurstmann"
    xsi:type="stu:graduatedPerson">
    <stu:friend>anna</stu:friend>
  </stu:student>
```

Solution - Exercise 2: XML, XML Schema

The corrected XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="studi-style.css"?>
<stu:studenten xmlns:stu="http://www.studivz.net/studiSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.studivz.net/studiSchema studivz.xsd">
  <stu:student
    date-of-birth="1980-07-01" gender="female" mail="anna@web.de"
    username="anna" xsi:type="stu:person">
    <stu:name firstname="Anna" lastname="Buttermann"/>
    <stu:friend>rudi</stu:friend>
    <stu:friend>peter</stu:friend>
  </stu:student>
  <stu:student date-of-birth="1980-07-04" gender="male" mail="peter@web.de"
    username="peter" xsi:type="stu:person">
    <stu:name firstname="Peter" lastname="Brotmann"/>
    <stu:friend>anna</stu:friend>
  </stu:student>
  <stu:student finalGrade="1.367" degree="Dipl. Ing." mail="rudi@web.de"
    date-of-birth="1950-07-08" username="rudi" gender="male"
    xsi:type="stu:graduatedPerson">
    <stu:name firstname="Rudi" lastname="Wurstmann"/>
    <stu:friend>anna</stu:friend>
  </stu:student>
</stu:studenten>
```

Create a Cascading Style Sheet (CSS) that renders the XML similarly to the following figure:



Solution - Exercise 3: XML, CSS

Example for a CSS file:

```
@CHARSET "UTF-8";
@namespace stu url("http://www.studivz.net/studiSchema");
stu|student:before {
  content: attr(username) " - " attr(mail);
  font-size: 1.5em;
}
stu|studenten {
  border: dotted;
  max-width: 400px;
  margin: 2em;
}
stu|name:before { content: attr(firstname) " " attr(lastname); }
stu|name{
  font-size: 1em;
  font-style: italic;
  display: block;
}
stu|student {
  margin: 2em;
  display: block;
}
stu|friend:before { content: "has Friend: " }
stu|friend { display: list-item; margin-left: 1.5em;}
```