

**Exercise – 5**  
**XML and Semantic Web Technologies – SoSe 2012**  
**Instructor: Prof. Dr. Dr. Lars Schmidt-Thieme**  
**Tutor: Umer Khan 29-05-2012**

---

Q1: Consider the following xml document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE movies [
<!ELEMENT movies (Movie+) >
<!ELEMENT Movie ( title, year, _director, (comment | newcomment)+)>
<!ATTLIST Movie id ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA) >
<!ELEMENT _director (#PCDATA)>
<!ATTLIST _director name CDATA #IMPLIED>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT newcomment (#PCDATA)>
<!ATTLIST comment lang CDATA #IMPLIED>
]>
<movies>
<Movie id="y56225">
<title>Love Story</title>
<year>1980</year>
<_director name="Coppola"/>
<comment lang=""/>
<newcomment >Oscar</newcomment>
<comment lang="de">1980 Warner Bros.</comment>
<!-- Famous movie of the 80s -->
</Movie>
</movies>
```

Create an equivalent XML Schema to this DTD, and add additional restrictions which could not be expressed with the DTD. (for example: enforce type of year etc)

Q2. a) Is the following type definition is valid for schema? Why or why not ?

```
<complexType name="listitem">
  <sequence>
    <all>
      <element name="title"></element>
      <element name="created" type="dateTime"></element>
      <element name="createdby"></element>
      <element name="dueTo" type="date"></element>
    </all>
    <sequence minOccurs="0" maxOccurs="unbounded">
      <element name="refer" type="anyURI"></element>
    </sequence>
  </sequence>
  <attribute name="id" type="anyURI"></attribute>
</complexType>
```

b) Is the following element declaration is valid for the purpose of the Unique Particle Attribution? Why not?

```
<xs:element name="names">
  <xs:choice maxOccurs="unbounded">
    <xs:element ref="first-name" />
    <xs:sequence>
      <xs:element ref="first-name" minOccurs="0" />
      <xs:element ref="middle-name" minOccurs="0" />
      <xs:element ref="last-name" />
    </xs:sequence>
  </xs:choice>
</xs:element>
```

c) Correct the element declaration from task 2(b), so that the following XML fragments can be generated:

Fragment 1 :

```
<first-name> Michael </first-name>
<middle-name> Joseph </middle-name>
<last-name> Jackson </last-name>
```

Fragment 2:

```
<first-name> Michael </first-name>
<last-name> Jackson </last-name>
```

Fragment 3:

```
<first-name> Michael </first-name>
```

Fragment 4:

```
<middle-name> Joseph </middle-name>
<last-name> Jackson </last-name>
```