



universität  
wien

## Klausur Interoperabilität

23. Juni 2014

18.30 – 19.30 Uhr

Workflow Systems and Technology Group

Fakultät für Informatik

Universität Wien

Univ.-Prof. Dr. Stefanie Rinderle-Ma

### Allgemeine Hinweise:

**Die Bearbeitungszeit beträgt 60 Minuten (1 Punkt entspricht daher einer Minute Bearbeitungszeit). Bitte benutzen Sie gegebenenfalls die auf den Aufgabenzetteln vorhandenen Tabellen und Vordrucke. Bitte schreiben Sie nicht mit Bleistift. Bitte versehen Sie jedes von Ihnen abzugebende Blatt mit Ihrem Namen und Matrikelnummer.**

**Name:**

**Vorname:**

**Matrikelnummer:**

**Bewertung:**

Aufgabe	1 (20 Punkte)	2 (20 Punkte)	3 (20 Punkte)

## Aufgabe 1: XML und Datenbanken (20 Punkte)

Betrachten Sie folgende SQL-Anweisungen:

```
CREATE TABLE Music (Title VARCHAR(20), Artist VARCHAR(20),  
Description XML);
```

```
INSERT INTO Music VALUES('Song1', 'John Smith',  
    '<?xml version="1.0" encoding="UTF-8"?>  
    <Description id="12">  
        <genre>classic</genre>  
        <year>1995</year>  
        <price>16.00</price>  
    </Description>');
```

a) Welche Ausgabe erzeugt folgende SQL Query

(4 Punkte)

```
SELECT * FROM Music;
```

b) Geben Sie das SQL/XML Statement (DB2 Syntax) an, das folgendes XML-Dokument erzeugt auf Basis des in Tabelle Music enthaltenen Tupels erzeugt:

(6 Punkte)

```
<?xml version="1.0" encoding="UTF-8"?>  
<Songs>  
    <Title>Song1</Title>  
    <Artist>John Smith</ Artist >  
    <Description id="12">  
        <genre>classic</genre>  
        <year>1995</year>  
        <price>16.00</price>  
    </Description>  
</Songs>
```

c) Gegeben sei folgendes XML-Fragment

```
<Songs>  
    <Title>Song1</Title>  
    <description>  
        <genre>classic</genre>  
        <year>1995</year>  
    </description>  
</Songs>
```

**c1) Bilden Sie das XML-Fragment unter Benutzung des EDGE-Ansatzes mit Inlining auf eine relationale Datenbanktabelle ab. Nutzen Sie dabei den folgenden Vordruck. (6 Punkte)**


**c2) Erklären Sie kurz den Unterschied zwischen EDGE-Ansatz mittels Inlining und mittels Value Tables (4 Punkte)**

## Aufgabe 2: Service-orientierte Architekturen (20 Punkte)

```
<definitions
  name="Interoperabilitaet"
  targetNamespace="urn:ns1"
  xmlns:wm="urn:ns1"
  xmlns:wmdata="urn:ns2"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

<types>
  <xsd:schema targetNamespace="urn:ns2">

    <xsd:complexType name="teams">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="id" type="wmdata:team"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="team">
      <xsd:sequence>
        <xsd:element name="country" type="wmdata:countryID"/>
        <xsd:element name="group" type="wmdata:groupID"/>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="captain" type="xsd:string"/>
        <xsd:element name="trainer" type="xsd:string"/>
        <xsd:element name="goalkeeper" type="xsd:string"/> <!-- torhueter -->
        <xsd:element name="defenders" type="wmdata:defenders"/> <!-- verteid. -->
        <xsd:element name="forwards" type="wmdata:forwards"/> <!-- stuermer -->
        <xsd:element name="games" type="wmdata:games"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:simpleType name="countryID">
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="\w{2}"/>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="groupID">
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="[ABCDEF]"/>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="defenders">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="defender" type="wmdata:player"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="forwards">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="forward" type="wmdata:player"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="games">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="game" type="wmdata:game"/>
      </xsd:sequence>
```

```

</xsd:complexType>

<xsd:complexType name="player">
  <xsd:all>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="age" type="xsd:integer"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="game">
  <xsd:all>
    <xsd:element name="date" type="xsd:date"/>
    <xsd:element name="time" type="xsd:time"/>
    <xsd:element name="city" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>

</xsd:schema>
</types>

<message name="group">
  <part name="group" type="wmdata:groupID"/>
</message>
<message name="teamlist">
  <part name="return" type="wmdata:teams"/>
</message>
<message name="team">
  <part name="id" type="wmdata:countryID"/>
</message>
<message name="games">
  <part name="response" type="wmdata:games"/>
</message>

<portType name="InteroperabilitaetPort">
  <operation name="getCountries">
    <input message="wm:group"/>
    <output message="wm:teamlist"/>
  </operation>
  <operation name="getGames">
    <input message="wm:team"/>
    <output message="wm:games"/>
  </operation>
</portType>

<binding
  name="InteroperabilitaetBinding"
  type="wm:InteroperabilitaetPort">
  ...
</binding>

<service name="InteroperabilitaetService">...</service>
</definitions>

```

**Welche Operationen sind für diese WSDL möglich. Schreiben Sie das Ergebnis in folgendem Schema auf (2 Punkte).**

```
RUECKGABE operation(attribut:typ, attribut:typ, ...)
```

Beispiel:

```
void main(vorname:string, adresse:adresstyp)
```

- a) Entwerfen Sie ein REST Service das dieselbe Funktionalität bietet wie ein durch die WSDL beschriebenes Service. Schreiben Sie nur die Ressourcen (Pfade) auf. Es sollen mindestens 2 Teams, mit je 2 Verteidigern und je 2 Stürmern genannt werden. (6 Punkte)**

Beispiel: Ein Wetter Service (so soll das Ergebnis aussehen, nur für die WM)

```
/wien  
/wien/hohe_warte  
/wien/hohe_warte/temperatur  
/wien/hohe_warte/luftdruck  
/vorarlberg  
/vorarlberg/pfaender  
/vorarlberg/pfaender/temperatur  
/vorarlberg/pfaender/luftdruck
```

- b) Beantworten Sie folgende Fragestellungen, jeweils im folgenden Schema (9 Punkte).**

```
OPERATION: Ressource  
RUECKGABE
```

Beispiel 1: Die Temperatur auf der hohen Warte?

```
GET: /wien/hohe_warte/temperatur  
17
```

Beispiel 2: Für welche Bundesländer gibt's Wetter Information?

```
GET: /wien/hohe_warte/temperatur  
<result><item>wien</item><item>vorarlberg</item></result>
```

Hinweis: Was zurückgegeben wird, wird meist durch Konvention bestimmt, das Datenformat (text, xml, ...) dürfen Sie frei wählen.

**C1) Welche Teams gibt's in Gruppe C?**

**C2) Das Team „Österreich“ soll in Gruppe A angelegt werden(mit Prohaska als Trainer, und Polster als Stürmer, 1 Spiel gegen Brasilien). (Hinweis: eine neue countryID wird entstehen).**

**C3) Verändern sie den Namen des Teams auf „Austria“.**

- c) Weshalb erstelle ich eventuell ein Web-Service, wenn ich alte Software in ein neues System integrieren soll? Wie gehe ich vor (Stichwort Zwiebel)? (3 Punkte)**

### **Aufgabe 3: Ontologien (20 Punkte)**

- a) Schreiben Sie eine Ontologie in Turtle Syntax nieder, welche EXACT die Zusammenhänge in der WSDL aus AUFGABE 2 ausdrückt. Bitte 2 Länder, je 2 Spiele, je 1 Stürmer (10 Punkte).
- b) Erstellen Sie eine SPARQL Abfrage: Alle Spiele (egal für welches Land), welche nach 17:59 Uhr und vor 20:01 Uhr beginnen (5 Punkte).
- c) Wann speichere ich Daten in Datenbanken, wann in XML Files, wann in Ontologien? Wie wirkt sich das auf die Sicherstellung der Interoperabilität aus. (5 Punkte)