



universität
wien

Klausur Interoperabilität

20. Juni 2013

08.00 – 09.00 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)

Gegeben sei folgende relationale Datenbank-Tabelle R. Wir nehmen an, dass die beiden Spalten A und B vom Typ XML (SQL/XML Standard) definiert sind, Attribut C jedoch vom Typ Integer.

R	A	B	C
	<A>Bakery	Ben	100
	<A>Butcher	Mike	200

a) Geben Sie das SQL/XML-Statement an, dass die obige Tabelle R erzeugt (ohne Werte!) (5 Punkte)

b) Skizzieren Sie, welche Ausgabe die folgende Anfrage erzeugt (5 Punkte)

```
SELECT A FROM R WHERE C=200;
```

c) Welche Ausgabe erzeugt folgende Anfrage? (5 Punkte)

```
SELECT XMLSERIALIZE (
  XMLDOCUMENT (
    XMLELEMENT ( NAME "R",
      XMLCONCAT (A, B)
    )
  )
  AS CLOB VERSION '1.0' INCLUDING XMLDECLARATION)
FROM R;
```

d) Welche Ausgabe erzeugt folgende Anfrage? (5 Punkte)

```
SELECT XMLSERIALIZE (
  XMLDOCUMENT (
    XMLELEMENT ( NAME "R",
      XMLAGG (A)
    )
  )
  AS CLOB VERSION '1.0' INCLUDING XMLDECLARATION)
FROM R;
```

Aufgabe 2: Service-orientierte Architekturen (20 Punkte)

```
<definitions
  name="WorkflowTechnologies"
  targetNamespace="urn:ns1"
  xmlns:kino="urn:ns1"
  xmlns:kdata="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="kinoliste">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="id" type="kdata:kino"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="kino">
      <xsd:sequence>
        <xsd:element name="id" type="kdata:kinoId"/>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="strasse" type="xsd:string"/>
        <xsd:element name="plz" type="xsd:string"/>
        <xsd:element name="bundesland" type="xsd:string"/>
        <xsd:element name="saele" type="kdata:saele"/>
        <xsd:element name="filme" type="kdata:filme"/>
        <xsd:element name="vorstellungen" type="kdata:vorstellungen"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:simpleType name="kinoId">
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="(K)\d{4}"/>
      </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="saele">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="saal" type="kdata:saal"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="filme">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="film" type="kdata:film"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="vorstellungen">
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="vorstellungen" type="kdata:vorstellung"/>
      </xsd:sequence>
    </xsd:complexType>
```

```

<xsd:complexType name="saal">
  <xsd:all>
    <xsd:element name="nummer" type="xsd:integer"/>
    <xsd:element name="plaetze" type="xsd:integer"/>
    <xsd:element name="reihen" type="xsd:integer"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="film">
  <xsd:all>
    <xsd:element name="titel" type="xsd:string"/>
    <xsd:element name="age" type="xsd:integer"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="vorstellung">
  <xsd:all>
    <xsd:element name="datum" type="xsd:date"/>
    <xsd:element name="uhrzeit" type="xsd:time"/>
    <xsd:element name="saal" type="xsd:integer"/>
  </xsd:all>
</xsd:complexType>

</xsd:schema>
</types>

<message name="where">
  <part name="country" type="xsd:string"/>
  <part name="city" type="xsd:string"/>
</message>
<message name="cinemalist">
  <part name="return" type="kdata:kinoliste"/>
</message>
<message name="cinema">
  <part name="id" type="kdata:kinoId"/>
</message>
<message name="cinemadetails">
  <part name="response" type="kdata:kino"/>
</message>

<portType name="WorkflowTechnologiesPort">
  <operation name="getCinemas">
    <input message="kino:where"/>
    <output message="kino:cinemalist"/>
  </operation>
  <operation name="getCinemaDetails">
    <input message="kino:cinema"/>
    <output message="kino:cinemadetails"/>
  </operation>
</portType>

<binding
  name="WorkflowTechnologiesBinding"
  type="kino:WorkflowTechnologiesPort">
  ...
</binding>

<service name="WorkflowTechnologiesService">...</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 wie 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 Kinos, mit je 2 Sälen und je 2 Filmen genannt werden. (6 Punkte)**

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

```
/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 Kinos gibt's in Wien?

C2) Das „Bellaria“ Kino in Wien soll angelegt werden(mit 1 Saal, und 1 Vorstellung, Filmtitel: „Der Rest ist Schweigen“). (Hinweis: eine neue kinolD wird entstehen).

C3) Verändern sie den Namen des Kinos auf „3D Mangler“.

- 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 1 Kino, 2 Filme, 1 Saal, 1 Vorstellung (10 Punkte).

Hinweis: eine Vorstellung verlinkt Filme und Säle – keine Strings.

- b) Erstellen Sie eine SPARQL Abfrage: Alle Filme (egal in welchem Kino), welche nach 17:59 Uhr und vor 20:01 Uhr beginnen (5 Punkte).
- c) Wie können Semantische Technologien genutzt werden um Interoperabilität zwischen System herzustellen? (5 Punkte)