

---

## Aufgaben zur Klausur in *Verteilte Systeme* (SS 2007)

Zeit: 90 Minuten,

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen auf gesonderten Blättern ein (Vorder- und Rückseite benutzen). **Vergessen Sie nicht, das Deckblatt zu unterschreiben.**

Für die Prüfung werden insgesamt 40 Bewertungseinheiten (BE) vergeben. Zum Bestehen benötigen Sie mindestens 20 BE. Die Verteilung der BE auf die einzelnen Aufgaben ist proportional zur erwarteten Ausführlichkeit der Antwort und der Zeit, die Sie sich damit beschäftigen sollten.

Viel Erfolg !

### 1. Aufgabe (4 BE), Thema: Allgemeine Anforderungen

- a) Erklären Sie, was der Begriff „transparent bzgl. einer bestimmten Eigenschaft“ nach ISO bedeutet! (1 BE)
- b) Betrachten Sie den Vorgang einer Notenabfrage über das Online-Sekretariat der FH Wedel: Erklären Sie, was die drei Transparenzeigenschaften *Ort*, *Ausfall* und *Skalierung* in diesem Zusammenhang bedeuten und geben Sie für jede ein Beispiel dafür an, was Ihr Browser Ihnen zeigen oder Sie fragen müsste, damit diese Eigenschaft *nicht* erfüllt ist. (3 BE)

### 2. Aufgabe (9 BE), Thema: Pheromonbasierte Verkehrssteuerung

Vergleichen Sie zwei dynamische Navigationssysteme in einem Fahrzeug, eines pheromonbasiert, das andere arbeitet mit einem zentralistischen Wegealgorithmus, z.B. Dijkstra oder A\*: Beide Systeme sollen mit topologisch identischen Straßenkarten arbeiten und in einem Zeitintervall von 30s die von ihnen benötigten Aktualisierungen von einer Zentrale abfragen dürfen. Beiden Systemen ist zu jedem Zeitpunkt der genaue Aufenthaltsort des Fahrzeugs und das gewünschte Ziel bekannt.

- a) Nennen und begründen Sie eine aus Sicht der Fahrzeugsoftware vorteilhafte Eigenschaft des pheromonbasierten Verfahrens! (4 BE)
- b) Skizzieren Sie grob, wie das pheromonbasierte System den in a) genannten Vorteil erreicht! (2 BE)
- c) Welchen Vorteil bietet die Verdunstung bei der natürlichen Ameisennavigation, und wie wird die Verdunstung beim ABC-Verfahren simuliert? (3 BE)

### 3. Aufgabe (6 BE), Thema: Grundlagen der Client-Server-Beziehung

- a) Erklären Sie die Begriffe *Protokoll* und *Mehrschichtenarchitektur*! (2 BE)
- b) Welchen Vorteil bietet die Verwendung einer Middleware zwischen Clients und Servern? (2 BE)
- c) Skizzieren Sie die Funktionsweise von *callback* und *polling* und nennen Sie ein Argument, das für *callback* und eines, das für *polling* spricht! (2 BE)

#### 4. Aufgabe (9 BE), Thema: Entfernte Aufrufe

- a) Welche Schwierigkeit ergibt sich für RMI, die sich bei RPC nicht stellt? (1 BE)
- b) Wie löst RMI diese Schwierigkeit? (2 BE)
- c) Ein Client definiert die Klasse `ClientClass` mit der Methode `call (s)`, welche die Methode `answer ()` der Klasse `ServerClass` mit der `ServerClass`-Instanz `s` aufruft: Beschreiben Sie die minimal notwendige RMI-Implementierung in Java! (Der Aufruf der Methoden von einem Hauptprogramm muss nicht realisiert werden. Benötigte Importe müssen nicht angegeben werden) (6 BE)

#### 5. Aufgabe (3 BE), Thema: Kommunikation über WebServices

- a) Wozu dient eine WSDL-Datei? (1 BE)
- b) Die von Axis generierte WSDL-Datei des Anhangs ist in sich nicht konsistent: Es wurde genau ein WSDL-Tag mit allen Unterelementen komplett gelöscht: Welches? (kurze Begründung!) (2 BE)

#### 6. Aufgabe (7 BE), Thema: SOA-Grundlagen

- a) Erläutern Sie an 2 Beispielen, dass das Konzept SOA und der Standard Web Services voneinander unabhängig sind, d.h. man kann eine SOA ohne WebServices realisieren und man kann mit WebServices andere Dinge realisieren als eine SOA. (2 BE)
- b) Nennen Sie eine Gemeinsamkeit und einen konzeptionellen Unterschied zwischen UDDI und WS Inspection! (2 BE)
- c) Erklären Sie den Unterschied zwischen funktionsorientierter und nachrichtenorientierter Datenübertragung! Welche Technik wird bei einer SOA gefordert? (3 BE)

#### 7. Aufgabe (2 BE), Thema: SOA for business

Welche Art von Problemen werden in EAI-Prozessen gelöst? (Übersetzung des Begriffs ist nicht ausreichend, da er zu schwammig ist)

## Anhang: WSDL-Schnittstelle

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:apachesoap=http://xml.apache.org/xml-soap
  xmlns:impl=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:intf=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:tns1="http://bean.tour9" xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/
  xmlns:wsdlsoap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <wsdl:types>
    <schema targetNamespace="http://bean.tour9" xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="Tour">
        <sequence>
          <element name="tripTourUnits" nillable="true"
            type="soapenc:ArrayOf_tns1_TripTourUnit" />
        </sequence>
        <attribute name="language" type="soapenc:string" />
      </complexType>
      <complexType name="HbtException">
        <sequence/>
      </complexType>
      <complexType name="UserException">
        <sequence/>
      </complexType>
    </schema>
  </wsdl:types>

  <wsdl:message name="calculateTourSplitSetRequest">
    <wsdl:part name="tour" type="tns1:Tour" />
  </wsdl:message>

  <wsdl:message name="calculateTourSplitSetResponse">
    <wsdl:part name="calculateTourSplitSetReturn" type="tns1:Tour" />
  </wsdl:message>

  <wsdl:message name="HbtException">
    <wsdl:part name="fault" type="tns1:HbtException" />
  </wsdl:message>

  <wsdl:message name="UserException">
    <wsdl:part name="fault" type="tns1:UserException" />
  </wsdl:message>

  <wsdl:message name="calculateTourRequest">
    <wsdl:part name="tour" type="tns1:Tour" />
  </wsdl:message>

  <wsdl:message name="calculateTourResponse">
    <wsdl:part name="calculateTourReturn" type="tns1:Tour" />
  </wsdl:message>

  <wsdl:portType name="TourSchedulingService">
    <wsdl:operation name="calculateTourSplitSet" parameterOrder="tour">
      <wsdl:input message="soapenc:calculateTourSplitSetRequest"
        name="calculateTourSplitSetRequest" />
      <wsdl:output message="soapenc:calculateTourSplitSetResponse"
        name="calculateTourSplitSetResponse" />
      <wsdl:fault message="soapenc:HbtException" name="HbtException" />
      <wsdl:fault message="soapenc:UserException" name="UserException" />
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

```

<wsdl:operation name="calculateTour" parameterOrder="tour">
  <wsdl:input message="soapenc:calculateTourRequest" name="calculateTourRequest"/>
  <wsdl:output message="soapenc:calculateTourResponse"
    name="calculateTourResponse"/>
  <wsdl:fault message="soapenc:HbtException" name="HbtException"/>
  <wsdl:fault message="soapenc:UserException" name="UserException"/>
</wsdl:operation>
</wsdl:portType>

<wsdl:binding name="TourSchedulingServiceSoapBinding"
  type="soapenc:TourSchedulingService">

  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="calculateTour">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="calculateTourRequest">
      <wsdlsoap:body encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        namespace="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="calculateTourResponse">
      <wsdlsoap:body encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        namespace="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"/>
    </wsdl:output>
    <wsdl:fault name="HbtException">
      <wsdlsoap:fault encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        name="HbtException" namespace=http://schemas.xmlsoap.org/soap/encoding/
        use="encoded"/>
    </wsdl:fault>
    <wsdl:fault name="UserException">
      <wsdlsoap:fault encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        name="UserException" namespace=http://schemas.xmlsoap.org/soap/encoding/
        use="encoded"/>
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>

<wsdl:service name="TourSchedulingServiceService">
  <wsdl:port binding="soapenc:TourSchedulingServiceSoapBinding"
    name="TourSchedulingService">
    <wsdlsoap:address location="http://localhost:8080/axis/services/">
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>

```