<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, April 13</td>
<td>Quiz corrections</td>
</tr>
<tr>
<td>Wednesday, April 21</td>
<td>Deadline for project reports</td>
</tr>
<tr>
<td></td>
<td>Project demos</td>
</tr>
</tbody>
</table>
Quiz #2 - Answers

INSE 7110 – Winter 2004
Value Added Services Engineering in Next Generation Networks
Week #10
Statistics ...

<table>
<thead>
<tr>
<th>Grade Range</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+ (30.5 – 35)</td>
<td>2</td>
<td>(Around 10%)</td>
</tr>
<tr>
<td>A- (26 – 30)</td>
<td>10</td>
<td>(Around 60%)</td>
</tr>
<tr>
<td>B+ (21.5 – 25.5)</td>
<td>5</td>
<td>(Around 30%)</td>
</tr>
</tbody>
</table>
Statistics ...

Questions for which the lowest marks were given:
- Questions for which the answers go a bit beyond professor’s lectures notes.
- Question 2 (Sequence diagrams)
- Equivalence of Parlay / Web services business roles and TINA-C business roles.
Question 2: Legacy based architectures …

Note:
1. There is a difference between re-using IN and inter-working with IN
2. The question was on re-using and not inter-working
3. Some “consolation” points were given to those who misunderstood
Communication between NGN switches and SCPs.

- Next generation switches do not support SS7
- INAP is ASN.1 based while some Internet Telephony protocols (e.g. SIP) are text based
Tentative solutions ...

Three main approaches

- First: Put the burden on the SCP side
  - IP transport
  - support of text based protocol (if SIP)

- Second: Put the burden on the NGN switches sides (e.g. support of SS7)

- Third: Gateways
The second issue ...

Call models

- IN call models were built explicitly for circuit switched telephony
- NGN “call models” were built without IN in mind
Tentative solutions

The call model issue: Two main approaches

- Integrated call model
- Call models (i.e. H.323/SIP and IN) running in parallel and interacting
Question #2 – Signaling protocol specific architectures

The two methods:
- DoInvite
- DoErrorResponse

Tolerated answers: Getheader, Setheader or more generally any pair of SIP servlet methods
Question #2 – DoInvite () Based – solution 1

A

Invite (B)

Proxy

Forward Invite

doInvite ()

Servlet container

B

C

OK

OK

ACK

ErrorResponse (Busy)

Getheader

Setheader

Invite (C)

OK
Question #2 – DoInvite () Based – solution 2

A                                     Proxy                     Servlet container                    B                                  C

Invite (B)                             →                             Invite (B)                             →                             Invite (B)                             →                             Invite (C)

Forward Invite                          ←                             ErrorResponse (Busy)                   ←                             Getheader

doInvite ()                             ←                             Setheader                              ←                             Invite (C)

OK                                      ←                             ACK                                    ←                             OK
Question #2 – DoErrorResponse () Based – solution

A                                Proxy                              Servlet container
              Invite (B)                       Invite (B)                       doInvite ()

              Forward Error Response

              Getheader

              Setheader

B                      ErrorResponse (Busy)                doErrorResponse ()

              Invite (C)

C                                OK                                OK

ACK
Question 3

- Roles
  - Client application
    - Consume/use the services (e.g. network capabilities)
    - Equivalent to end users in TINA-C.
  - Enterprise operator
    - The entity that subscribes to the services
    - Subscriber in TINA-C
  - Framework operator
    - Entity that handles the subscriptions
    - Equivalent to the retailer in TINA-C
Question 3

– Two types of APIs
– Services APIs
  • Expose the network capabilities (e.g. call control, presence)
– Framework APIs
  • Make the use of the service APIs secure, accountable and resilient (e.g. security, registration, authentication)
Question 3

Generic call control service

Multiparty call control service

Multimedia call control service

Conference call control service
Specificities:
- Only architecture that aims at service creation by end-users

Prime target: Un-trusted parties
- Direct use
- Use via a graphical user interface
  • Higher level of abstraction
  • Mapping done by middleware
Question 3

Lightweight, efficient easy to implement

Easily verifiable for correctness

Executable in a safe manner

Easily writeable and parsable

Extensible

Signaling protocol independence
Question 4

- Extensible Mark Up Language (XML)
- Web Service Description Language (WSDL)
- Simple Object Access Protocol (SOAP)
- Universal Description Discovery and Integration (UDDI)
- Putting it together
Question 4

Broker (Human + agent)

Requestor (Human + agent)

Provider (Human + agent)

TINAC Broker

TINA_C Consumer/End-user

TINA_C Retailer
Question 4

The adapter pattern
Question 4

The gateway pattern
Question 4

The proxy pattern
Question 4

The delegate pattern
Question 4

The orchestrator pattern
Question 4

The filter pattern

- Requestor
- Filter
- Web Service 1
- Web Service 1
- Web Service 1

1 2 3
1 2 3
Question 4

The workflow pattern

Requestor 1/ Web service 4

Requestor 2/ Web service 1

Requestor 3/ Web service 2

Requestor 4/ Web service 3

1 2 3 4
Question 4  ...

Service locators

- Interact (on behalf of service requestor with UDDI and/or catalogues to find service(s) meeting specific criteria
- May be deployed by providers to direct to her/his services
- May be deployed by an independent party
- Accessible via a standardized API

- Catalogues
  - Standardized way for service providers to provide more details about their services (e.g. closing hours of an outlet)
  - Kept in service provider domain
  - Accessible via a standardized API by:
    - Service requestors
    - Service locators