Chapter II – Addendum
(Stepwise procedure for REST modelling)

Case Study – REST for Conferencing

http://users.icens.concordia.ca/~glitho/
References

Case Study On Conferencing

1. A stepwise procedure
2. On conferencing semantics
3. Applying the procedure to conferencing
The procedure – First Part

- Figure out the data set
- Split the data set into resources
The procedure – Second Part

For each resource:
- Name the resources with URIs
- Identify the subset of the uniform interface that is exposed by the resource
- Design the representation(s) as received (in a request) from and sent (in a reply) to the client
- Consider the typical course of events by exploring and defining how the new service behaves and what happens during a successful execution
On Conferencing semantics

- The conversational exchange of multimedia content between several parties
  - About multimedia
    - Audio, video, data, messaging
  - About participants
    - Any one who wants to participates the conference
On Conferencing semantics

Classification:
- Dial-in / dial-out
- Open/close
- Pre-arranged/ad hoc
- With/without sub-conferencing (i.e. sidebar)
- With/without floor control
On conferencing semantics

- Case considered in the use case
  - Create a service that allows a conference manager to:
    - Create a conference
    - Terminate a conference
    - Get a conference status
    - Add users to a conference
    - Remove users from a conference
    - Change media for a participant
    - Get a participant media
Applying the procedure – First part

1. Data set

- Conferences
- Participants
- Media
Applying the procedure – First part

2. Split the data set into resources
   - Each conference is a resource
   - Each participant is a resource
   - One special resource that lists the participants
   - One special resource that lists the conferences (if we consider simultaneous conferences)
Applying the procedure – Second part

3. Name the resources with URIs
   - I’ll root the web service at
     http://www.confexample.com/
   - I will put the list of conferences at the root URI
   - Each conference is defined by its ID:
     http://www.confexample.com/{confId}/
   - A conference participants’ resources are subordinates of the conference resource:
     - The lists of participants:
       http://www.confexample.com/{confId}/participants/
     - Each participant is identified by his/her URI:
       http://www.confexample.com/{confId}/participants/{participantURI}/
## Applying the procedure – Second part

<table>
<thead>
<tr>
<th>Resource</th>
<th>Exposed subset of the uniform interface</th>
<th>Data representation operation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operation</td>
<td>Client-&gt;server</td>
</tr>
<tr>
<td></td>
<td><strong>HTTP action</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Conference                | Create: establish a conference          | POST: http://confexample.com/| <conference>
|                           |                                        | <description> discuss project <description>
|                           |                                        | <maxParticipants>10</maxParticipants>
|                           |                                        | http://www.confexample/conf23@example.com |
|                           | Read: Get conference status             | GET: http://confexample.com/{confid} | None |
|                           |                                        | <status>Active</status>      |                              |
|                           | Delete: end a conference                | DELETE: http://confexample.com/{confid} | None |
|                           | Read: Get list of participants          | GET: http://confexample.com/{confid}/participants | None |
| List of participant(s)    | Create: Add a participant              | POST: http://confexample.com/{confid}/participants | <participant>
|                           |                                        | alice@ericsson.com          | <participant> |
|                           |                                        |                               | <status>invited</status>     |
|                           | Read: Get a participant status          | GET: http://confexample.com/{confid}/participants/{participantURI} | None |
|                           | Delete: remove a participant            | DELETE: http://confexample.com/{confid}/participants/{participantURI} | None |
Applying the procedure – Second part

Diagram:
- Alice (REST client)
- Conf app (REST server)
- Bob

Steps:
1. POST (http://www.confexample.com)
2. 202 accepted (http://www.confexample.com/conf1@confexample.com)
3. 200 OK
4. GET (http://www.confexample.com/conf1@confexample.com)
5. 200 OK
6. POST (http://www.confexample.com/conf1@confexample.com/participants, bob@ericsson.com)
7. 202 accepted
8. INVITE
9. OK
10. ACK
11. 200 OK

Notes:
- The server creates the conference
- The server adds the participant(s) to the conference
Applying the procedure – Second part

9. What might go wrong?
  - Conference

<table>
<thead>
<tr>
<th>Operation</th>
<th>Server-&gt;Client</th>
<th>Way it may go wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create (POST)</td>
<td>Success: 200 OK</td>
<td>The received request is not correct (e.g. has a wrong body)</td>
</tr>
<tr>
<td></td>
<td>Failure: 400 Bad Request</td>
<td></td>
</tr>
<tr>
<td>Read (GET)</td>
<td>Success: 200 OK</td>
<td>The targeted conference does not exist</td>
</tr>
<tr>
<td></td>
<td>Failure: 404 Not Found</td>
<td></td>
</tr>
<tr>
<td>Delete (DELETE)</td>
<td>Success: 200 OK</td>
<td>The targeted conference does not exist</td>
</tr>
<tr>
<td></td>
<td>Failure: 404 Not Found</td>
<td></td>
</tr>
</tbody>
</table>
Applying the procedure – Second part

9. What might go wrong?
   ▪ Participant(s)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Server-&gt;Client</th>
<th>Way it may go wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create (POST)</td>
<td>Success: 200 OK Failure: 400 Bad Request Failure: 404 Not Found</td>
<td>• The received request is not correct (e.g. has a wrong body)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The target conference does not exist</td>
</tr>
<tr>
<td>Read (GET)</td>
<td>Success: 200 OK Failure: 404 Not Found</td>
<td>• The target conference does not exist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The target participant does not exist</td>
</tr>
<tr>
<td>Update (PUT)</td>
<td>Success: 200 OK Failure: 400 Bad Request Failure: 404 Not Found</td>
<td>• The received request is not correct</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The target conference does not exist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The target participant does not exist</td>
</tr>
<tr>
<td>Delete (DELETE)</td>
<td>Success: 200 OK Failure: 404 Not Found</td>
<td>• The target conference does not exist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The target participant does not exist</td>
</tr>
</tbody>
</table>
Telecommunication Services Engineering (TSE) Lab