Content Adaptation: Application Scenarios

T-110.456 Next Generation Cellular Networks
2005-04-20
Oskari Westerholm
Agenda

• Introduction
• Scenario for Content Selection: Browsing
• Scenario for Transcoding: MMS
• Real-Life Examples
• Conclusions
# Introduction:

## Types and Methods of Content Adaptation

(as seen on previous lectures)

<table>
<thead>
<tr>
<th>Types</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• format adaptation</td>
<td>• multimedia transcoding</td>
</tr>
<tr>
<td>• characteristics adaptation</td>
<td>• content selection</td>
</tr>
<tr>
<td>• appearance adaptation</td>
<td>• rendering at the client</td>
</tr>
<tr>
<td>• encapsulation adaptation</td>
<td>• hybrid approaches</td>
</tr>
<tr>
<td>• message size adaptation</td>
<td></td>
</tr>
</tbody>
</table>

(may include any or all of above)
Scenario for Content Selection: Browsing (1/4)

1. client requests an *abstract* URL and tells its capabilities
2. server resolves capabilities, may also use databases
3. server selects contents (algorithm follows)
4. server may do some transcoding or XSLT operations
5. server delivers the content
Browsing (2/4): Content Selection Algorithm

Production of multimedia content descriptors:
  for each multimedia element
    - set requirements and value for each version
    - sort versions by value

Content selection:
  for each requested element
    - start the search from version of maximum value
    - move to lower value until all requirements are met
    - deliver this version of the element
Browsing (3/4): Example Descriptors

• Multimedia Content Descriptors
  – Value
  – MinBitRate
  – MinImageResolution
  – MinVirtualScreenSize
  – MediaFormat

• Media Capability Descriptors
  – AvgBitRate
  – MaxImageResolution
  – VirtualScreenSize
  – MediaFormatSet

<table>
<thead>
<tr>
<th>Terminal</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitRate</td>
<td>15000</td>
<td>15000</td>
<td>50000</td>
</tr>
<tr>
<td>MaxImageResolution</td>
<td>320×240</td>
<td>50×50</td>
<td>320×240</td>
</tr>
<tr>
<td>VirtualScreenSize</td>
<td>320×480</td>
<td>80×80</td>
<td>320×240</td>
</tr>
<tr>
<td>MediaFormatSet</td>
<td>&quot;image/jpeg&quot;, &quot;image/gif&quot;, &quot;image/vnd.wap.wbmp&quot;</td>
<td>&quot;image/jpeg&quot;, &quot;image/gif&quot;, &quot;image/vnd.wap.wbmp&quot;</td>
<td>&quot;image/jpeg&quot;, &quot;image/gif&quot;, &quot;image/vnd.wap.wbmp&quot;, &quot;video/3gp&quot;</td>
</tr>
</tbody>
</table>

Received content

Note:
- BitRate too low for receiving utility 5 image.
- VirtualScreenSize too low for utility 4 image.
- A BitRate lower than 43200 would have resulted in utility 5 image.

Figure 7.16 Selected media content for different browsing devices.
Browsing (4/4): Adaptation Results

Device 1
Finland
Mo Jun 23rd
20°C
Wind SW 10 km/h

Device 2
Finland
20°C

Device 3
Monday, Jun 23rd
Finland: Sunny, 20°C, Wind SW 1
Scenario for Transcoding: MMS

- MT & NW evolution in progress => transcoding of messages needed
- MMS standardization & cooperation (e.g. platform independence)
- dynamic UAProf helps recognize additional installed software
- person vs. application originated content: transcoding vs. content selection
Real-Life Examples

- Equipment used: Nokia 6630 (Series 60)
  - standard browser
  - Opera for mobile, including Small Screen Rendering

- std browser
- std plaza
- timeout after 300 kB ;)
- scroll mode and narrow screen mode
- Opera in scroll mode
- wap plaza
- different URL typed by user
- two tabs
- Opera in SSR mode, using narrow screen
  - not just narrow layout, but some elements left out completely
  - some images left out by default
  - tries to find the essential information
Real-Life Examples: Bus Schedules (1/4)

- standard page
- pda version (by user request)
Real-Life Examples: Bus Schedules (2/4)

- std browser in narrow screen mode
- std page
- looks quite good, if you remember the meaning of each text line
- ~30 kB, quite fast

- std browser in narrow screen mode
- pda page (content selection at the server)
- easier to understand
- even faster
- map images different than on the regular page, designed for mobile screens
Real-Life Examples: Bus Schedules (3/4)

- Opera in scroll mode
  - std page
  - somewhat usable if you like to scroll around
- Opera in SSR mode
  - std page
  - very confusing
  - => SSR quite useless here
- Opera in scroll mode
  - pda page
  - very informative, no need to scroll
- Opera in SSR mode
  - pda page
  - about as usable as the previous example
Real-Life Examples: Bus Schedules (4/4)

- caching & rendering at the terminal:
- user selects bus stops and bus lines, terminal fetches schedules from the web
- data cached along with expiry date
- very usable, very fast
Conclusions

• complex issues, no single solution
• standardization going on

• from the examples:
  – no automatic identification of mobile users
  – delivery time not always the main issue
  – content selection vs. rendering at terminal give contradictory results => both need development
  – DIY also an option