Content Adaptation For Mobile Internet Application Scenarios

Sateesh Babu, Kavuri
Content Adaptation – Whole Picture
Motivation For Adaptation

- Pervasive Computing on the rise
- Varying terminal capabilities – WLAN, GPRS, UMTS etc.
- Varying network capabilities
- User preferences – timely delivery, no graphics/flashy graphics
- The evolved concept – Content Adaptation

I have a UMTS based mobile phone and I would like a video clip of the fashion show.

I have a GSM phone. I would want just the current score.
**Methods Of Adaptation**

**Multimedia Transcoding**

- **Capability Negotiation**
- **Capability Database**
- **Adaptation Policies**
- **Media Adaptation Engine**

---

**Architecture Of a Media Transcoder System**

- **Application-Specific Controller**
  - **Capability Negotiation**
    - Extracts capabilities info from protocol. E.g. -UserAgent Header -UAProf -Etc.
  - **Capability DB**
    - Contains the capabilities of different terminals useful for a wide range of applications
  - **Adaptation Policies**
    - Decide how the content should be adapted (drive the media adaptation modules)
  - **Media Adaptation Eng.**
    - Apply transformation to the media content or select the right version.

---

**Policies Plugins**
- MMS
- Browsing
- SIP IM
- Rich Calls

**Media Plugins**
- GIF
- JPEG
- WML
- H.263

---

**Requests/Responses**
Content Selection

- Multiple version of the content are stored
- Server selects the best version that fits the terminal capabilities
- Based on two key technologies:
  - Infopyramid
    - Provides a multimodal, multiresolution representation hierarchy for multimedia content.
  - Customizer
    - Customises content for a target terminal

Info Pyramid Of a Video Item
Adaptation Architectures

(a) Adaptation at source

(b) Adaptation at intermediary

(c) Adaptation at destination
Application Scenario

1. Client requests content of URL to server and provides its capabilities (UA header and optionally UAProf).
2. Server resolves UAProf capabilities and possibly gets additional capabilities from a local database, if needed, using UA header or static UAProf URL.
3. Server selects the best content according to terminal capabilities and its content selection policies.
4. Server may perform additional transcoding or XSLT operations.
5. Server delivers the adapted content to the client.
Content Selection Algorithm

• Media content tagged with set of requirements – “Multimedia Content Descriptors” (MCDs).
• MCDs need to be fulfilled by the terminal, the network and the user preferences.
• Algorithm based on a comparison between multimedia content descriptors and capabilities and characteristic descriptors.

<table>
<thead>
<tr>
<th>Production of multimedia content descriptors:</th>
</tr>
</thead>
<tbody>
<tr>
<td>for each multimedia element</td>
</tr>
<tr>
<td>Set the requirements for each version of multimedia element (usually done under the</td>
</tr>
<tr>
<td>author’s supervision)</td>
</tr>
<tr>
<td>Order the version in increasing order of value or quality</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Content selection:</th>
</tr>
</thead>
<tbody>
<tr>
<td>for each requested element (WML deck, (X)HTML page, inline image, audio, video, etc.)</td>
</tr>
<tr>
<td>Select the first element in the list of versions for which all the requirements are</td>
</tr>
<tr>
<td>satisfied; thus the search for a match starts from the version with highest value to the least value until a match occurs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return request:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the selected version of the element (WML deck, (X)HTML page, inline image,</td>
</tr>
<tr>
<td>audio, video, etc.) to the requesting entity</td>
</tr>
</tbody>
</table>
### Sample Multimedia Content Descriptors

#### (for a weather service)

<table>
<thead>
<tr>
<th>Versions</th>
<th>Usage</th>
<th>Context</th>
<th>Requirements</th>
<th>Sunny</th>
<th>Sunny</th>
<th>Sun</th>
<th>Sunny</th>
<th>Sunny</th>
<th>Sunny</th>
<th>Sunny</th>
<th>Sunny</th>
<th>Sunny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility (Value)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BitsPerPixel</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MinBitRate (bps)</td>
<td>9600</td>
<td>300</td>
<td>9600</td>
<td>9600</td>
<td>28800</td>
<td>43200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MinImageResolution</td>
<td>0x0</td>
<td>0x0</td>
<td>16x16</td>
<td>48x48</td>
<td>180x180</td>
<td>128x128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MinVirtualScreenSize</td>
<td>0x0</td>
<td>50x50</td>
<td>50x50</td>
<td>100x80</td>
<td>200x200</td>
<td>160x160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MediaFormat</td>
<td>audio/amr</td>
<td>text/plain</td>
<td>image/vnd.wap.wbmp</td>
<td>image/gif</td>
<td>image/jpeg</td>
<td>Video/3gp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Media Capability Descriptors

- **Utility (Value)**: a positive integer setting the rank of this version with respect to other related versions (where 1 is the order of the image having the lowest value).
- **MinBitRate**: minimum required bit rate in bits/s (bps).
- **MinImageResolution**: minimum image resolution required (X Y pixels).
- **MinVirtualScreenSize**: the minimum virtual screen size (X Y pixels) under which the image should be displayed.
- **MediaFormat**: the media format in which the picture is stored. Should be in a terminal acceptable format.
Terminal Media Capability Descriptors

- Associated with those multimedia content descriptors, the terminal would provide its media capability descriptors (MCD) when making a request for content.
- BitRate: the terminal’s connection average bit rate.
- MaxImageResolution: the terminal’s maximum image resolution supported (X Y pixels).
- VirtualScreenSize: the terminal’s virtual screen size (X Y pixels).
- MediaFormatSet: the terminal’s supported media formats.
## Results of Media Content Selection

<table>
<thead>
<tr>
<th>Terminal</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BitRate</strong></td>
<td>15000</td>
<td>15000</td>
<td>50000</td>
</tr>
<tr>
<td><strong>Maximum Image Resolution</strong></td>
<td>320x240</td>
<td>50x50</td>
<td>320x240</td>
</tr>
<tr>
<td><strong>Received Content</strong></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
</tbody>
</table>

**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.

---

**Finland**
- Tue Nov 15
- 20°C
- Wind SW 10km/h

**Device 1**

**Finland**
- 20°C

**Device 2**

**Tuesday, Nov 15**
- Finland, Sunny, 20°C, Wind SW 10km/h

**Device 3**
Multimedia Messaging Service - An evolutionary service to SMS
Can be used to deliver rich content – images, video, audio, music
An important 3G enabler
Architecture and concepts standardized by 3GPP – currently under OMA
Server-side multimedia message adaptation (MMA) – attempt to reduce interoperability problems.
MMA involves adapting content in the “Multimedia Messaging Service Center” (MMSC)
MMS Conformance Document – Ensuring some degree of terminal interoperability
MMS Applications

• MMS – provides generic mechanism to encapsulate and transport multimedia content.

• MMS provided applications:
  • Mobile to mobile: sending/receiving photos, audio/video clips, voicemail, business cards, and so on.
  • Web applications to mobile devices: electronic postcards, greeting cards, advertisement, news of the day (video/audio clips), screen savers, animations, maps.
  • Internet to/from mobile devices; receive selected emails, send emails.
  • Enabler for interactive games
MMS Transactions

MMS transaction and adaptation framework
1. The sender's terminal initiates a WAP POST (using WSP or HTTP) request to the MMSC. This operation uploads the message to the MMSC. The MMSC is then responsible for the delivery.

2. After the MMSC has stored the message, it sends a notification to the message recipient's terminal to inform it that a new message arrived. The notification is typically carried using WAP PUSH (e.g., SMS as the bearer). The notification contains a URL associated with the message. It also contains information about the message such as when the message expires, the message size, and optionally the sender's address.

3. The notification triggers in the recipient's terminal a WAP GET (using WSP or HTTP) operation that fetches the message (using its URL) from the MMSC to the mobile device. That transaction contains information about the terminal type (UA header) and may contain information about the terminal capabilities using UAPProf.

4. The MMSC retrieves, from its database, the message corresponding to the URL. It then may adapt the message to meet the terminal capabilities;

5. The MMSC sends the resulting message to the destination terminal.

6. The terminal confirms reception of the message.

7. The MMSC may send a delivery report to the sender using WAP PUSH.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MmsMaxMessageSize</td>
<td>The maximum size of a multimedia message in bytes.</td>
</tr>
<tr>
<td>MmsMaxImageResolution</td>
<td>The maximum size of an image in units of pixels (horizontal/vertical)</td>
</tr>
<tr>
<td>MmsCcppAccept</td>
<td>List of supported content types conveyed as MIME types</td>
</tr>
<tr>
<td>MmsCcppAcceptCharset</td>
<td>List of character sets that the MMS client supports; each item in the list is a character set name registered with IANA</td>
</tr>
<tr>
<td>MmsCcppAcceptLanguage</td>
<td>List of preferred languages; the first item in the list should be considered the user’s first choice; property value is a list of natural languages, where each item in the list is the name of a language as defined by (IETF RFC 1766)</td>
</tr>
<tr>
<td>MmsCcppAcceptEncoding</td>
<td>List of transfer encodings that the MMS client supports; property value is a list of transfer encodings, where each item in the list is a transfer encoding name as specified by (RFC 2045) and registered with IANA</td>
</tr>
<tr>
<td>MmsVersion</td>
<td>The MMS versions supported by the MMS client conveyed as majorVersionNumber.minor VersionNumber</td>
</tr>
<tr>
<td>MmsCcppStreamingCapable</td>
<td>Indicates whether the MMS client is capable of invoking streaming</td>
</tr>
<tr>
<td></td>
<td>(introduced in MMS 1.1)</td>
</tr>
</tbody>
</table>
References

• Content Networking in the Mobile Internet, editor(s): Sudhir Dixit, Tao Wu

• http://www.openmobilealliance.org/

• http://www.w3.org/


