MORE: A Mobile Open Rich Media Environment

Vidya Setlur     Tolga Capin      Suresh Chitturi
Ramakrishna Vedantham     Michael Ingrassia

Nokia Research Center
Palo Alto, California
Rich Media - Some Background

Rich media content is referred to dynamic content that is graphically rich and contains compound or multiple media, including graphics, text, video and audio, delivered through a single interface.

Providing a unique, richer and more compelling mobile consumer experience,

Rich media is currently a work item in both 3GPP and OMA.
Related Work

Macromedia Flash

MPEG-4 LASeR
System Architecture
Rich Media Type

- Rich media content (scene) is dynamically updated (updates) with small changes rather than a complete document being sent every time.

- A **scene**: Spatial organization of scene elements, the temporal organization of scene elements, synchronization information, and interaction among the elements. A scene is a representation of a fully complaint SVG document that may or have been updated over time.

- **Scene updates**: Incremental updates to the SVG Micro Document Object Model (uDOM). Updates include element **addition**, element **deletion**; element **replacement** and element attribute updates. Element replacement can even be used to replace an entire scene.
Initial Scene with a rectangle element.

Scene Update with a ‘DOMAttrModified’ operation.

Scene Update delivery with a 'DOMNodeInserted' operation.

Transmission of Scene Update delivery with a 'DOMNodeRemoved' operation.

Transmission of Scene Update with a ‘DOMNodeReplaced’ operation.

Transmission of Scene Update with a ‘group’ of updates with circle appended and ellipse removed.
Temporal Management of Scenes and Scene Updates

- Every scene and scene update sample is associated with a timestamp that refers to the time at which the scene/scene update is to be rendered on the client. The resolution of this timestamp ($T_{sfreq}$) is defined by the content creator.

- For example, the timestamp of the first scene is $T_{s1}$ and it is rendered at time $T_{presStart}$. If a succeeding scene update sample has a time stamp of $T_{su1}$, it is rendered at time $T_{presStart} + (T_{su1} - T_{s1})/T_{sfreq}$. 
Abbreviations:
S – Scene
SU – Scene Update
TS - Timestamp

Example: Rendering time of S(TS=600)

\[ T_{\text{presStart}} = 10:30:00, T_{S0} = 0, T_{S1} = 600 \]

\[ T_{\text{Sfreq}} = 10\text{Hz} \]

Rendering time = \( T_{\text{presStart}} + \frac{(T_{S1} - T_{S0})}{T_{\text{Sfreq}}} \)

\[ = 10:30:00 + 60\text{s} = 10:31:00 \]
Container Format

- Scenes are carried in scene tracks in ISO family files. They therefore use:
  - a video media handler ‘vmhd’;
  - a media handler type of ‘sdsms’ (scene description media handler);
  - a derivative of the base SampleEntry in the sample description box.
- The timescale for the rich media stream should be suitably chosen to achieve the desired accuracy of timing of access units.
- Other resources may be carried in the meta-data directories of ISO files.
- Support for sync samples for tune-in, interleaving, time synchronization.
Sample Entry Format

class SceneConfiguration extends FullBox ('dimC'){
    unsigned int(8) profile;
    unsigned int(8) level;
    string content_encoding;
    string text_encoding;
    string content_script_type;
}

class MPEG4BitRateBox extends Box('btrt'){
    unsigned int(32) bufferSizeDB;
    unsigned int(32) maxBitrate;
    unsigned int(32) avgBitrate;
}

class DIMSSampleEntry() extends SampleEntry ('dims'){
    SceneConfiguration config;
    MPEG4BitRateBox ();
}
## Transport Mechanisms

<table>
<thead>
<tr>
<th>Unicast</th>
<th>Broadcast/Multicast</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Streaming</strong></td>
<td></td>
</tr>
<tr>
<td>Video</td>
<td>Video</td>
</tr>
<tr>
<td>Audio</td>
<td>Audio</td>
</tr>
<tr>
<td>Timed Text</td>
<td>Timed Text</td>
</tr>
<tr>
<td>SVG</td>
<td>3GP file format</td>
</tr>
<tr>
<td>RTP Payload Formats</td>
<td>3GP file format</td>
</tr>
<tr>
<td>RTP/UDP</td>
<td>3GP file format</td>
</tr>
<tr>
<td><strong>Download</strong></td>
<td></td>
</tr>
<tr>
<td>Capability Exchange</td>
<td>Video</td>
</tr>
<tr>
<td>Scene Description</td>
<td>Audio</td>
</tr>
<tr>
<td>Presentation Description</td>
<td>Timed Text</td>
</tr>
<tr>
<td>Description</td>
<td>Text</td>
</tr>
<tr>
<td>Bitmap Graphics</td>
<td>Text</td>
</tr>
<tr>
<td><strong>Streaming</strong></td>
<td></td>
</tr>
<tr>
<td>SVG</td>
<td>FLUTE (with FEC)</td>
</tr>
<tr>
<td>RTP Payload Formats</td>
<td></td>
</tr>
<tr>
<td>RTP/UDP</td>
<td></td>
</tr>
<tr>
<td>IP (Unicast)</td>
<td></td>
</tr>
<tr>
<td>IP (Multicast)</td>
<td></td>
</tr>
</tbody>
</table>
Transport Mechanisms

RTP Header

<table>
<thead>
<tr>
<th>V=2</th>
<th>P</th>
<th>X</th>
<th>CC</th>
<th>M</th>
<th>PT</th>
<th>sequence number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>timestamp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>synchronization source (SSRC) identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>contributing source (CSRC) identifiers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>....</td>
</tr>
</tbody>
</table>

Common Payload Header

<table>
<thead>
<tr>
<th>T</th>
<th>A</th>
<th>P</th>
<th>CTR</th>
<th>R</th>
</tr>
</thead>
</table>

Payload Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Single DIMS unit</td>
</tr>
<tr>
<td>1</td>
<td>Aggregation Packet</td>
</tr>
<tr>
<td>2</td>
<td>Fragmentation Packet</td>
</tr>
</tbody>
</table>
<metadata>; MSG_ID=1;ELEMENT_ID="my-button1"; EVENT="click";[OCTET1.OCTET2. ... OCTETN];
MORE Client

Application Layer
- Rich Media Enabled User Agent
- Rich Media Enabled Browser

Services Layer
- Rich Media Client API
  - SVG Mobile 1.2 Engine
    - Synchronization Module
  - Audio/Video Module
  - Remote Interaction

Transport Layer
- Media Depacketizers
- Unicast and Broadcast Transport Protocols
Example
Reference Model Analysis (Ref: Rogge et al.)

- **Temporal Granularity**: The clock values for MORE have the same syntax as in SMIL (Synchronized Multimedia Integration Language) animation. The presentation time indicates the overall master timeline. Animations as well as embedded continuous media can have start and end times,

- **Interaction**: MORE provides additional functionality than just simple navigational interaction between documents and local media control interaction.

- **Extensibility**: MORE is based on open standards, and is not tightly bound to a particular solution, for example compression, allowing the flexibility to accommodate more optimal methods if available.

- **Reusability**: MORE supports reusability of media elements, content fragments and documents by inheriting this functionality from SMIL.

- **Adaptability**: MORE supports dynamic adaptation to preferences by using the `<switch>` tag borrowed from SMIL. Also, packet size and error concealment schemes can be adapted based on the network conditions, priority of content and the application.

- **Presentation-Neutral Representation**: Using SVG as the primary presentation format in MORE, a complete presentation description can be provided, independent of the rich media player.
Thanks

Reviewers and organizers of ICME 2006
Nokia Research Center Graphics and Vision Group