

A declarative approach to Broadcast TV

Jean-Charles Verdié, MStar Semiconductors Inc.
jc.verdie@mstarsemi.com

In a vision of a pure web-based UI & OTT for TV, many competing standards arise. However all of them work with the assumption that the web browser will sit on top of a full fledged operating system/middleware and will have plug-in or javascript object - based interaction, such as video broadcast object from OIPF, or channel list.

This turns the presentation engine into a procedural effort while the channel list is declarative per se, and therefore implies:

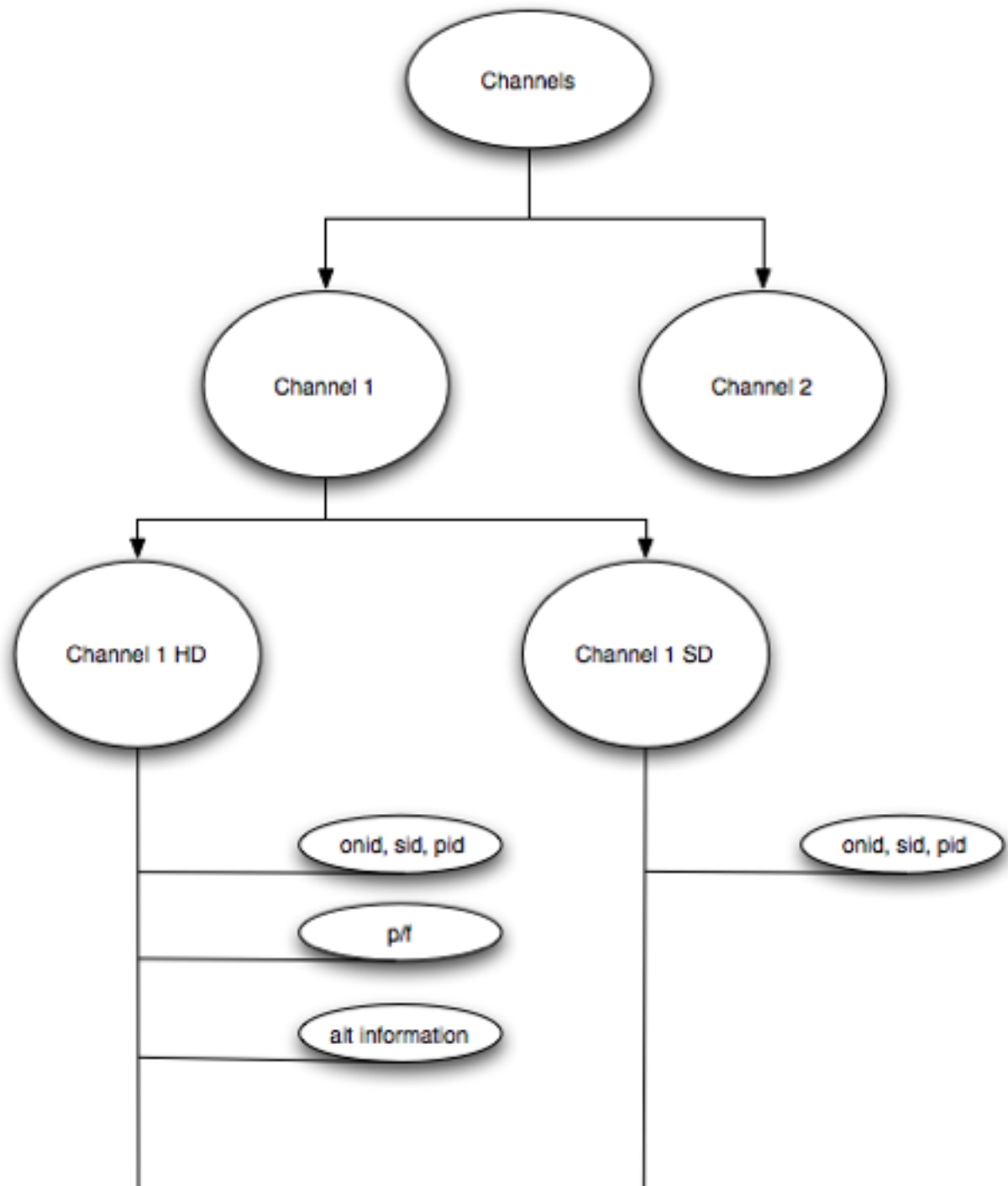
- more programmatic effort to achieve on the web application side
- binding constraints between JavaScript object / plugin and middleware

We propose to reassess the level of interaction needed between web application and middleware with the integration of middleware information into the DOM as an XML structure.

This proposition decouples data storage and access from its construction and processing

In order to achieve this goal we wish to introduce the notion of "channel" in XML content

Just like chat or contact have been formalized, we propose a standard way to operate DVB-triplet (or equivalent) and interact with it rather than just exposing a database-backed object with API-level interaction



That formalization would allow DOM-Level interaction with these objects, such as:

- next program, previous
- favorite management
- tuner scan
- EPG access

This proposition creates an obvious and ubiquitous way to integrate data from heterogeneous sources such as DVB, ATSC, IP & others, handling all of them under the same paradigm.

Engine parsers should understand that type of page as an informational page for tv

```
<?xml version="1.0"?>
<webtv xmlns='http://www.w3.org/2011/webtv'
        xmlns:dvb='http://www.w3.org/2011/webtvdvb'
        xmlns:iptv='http://www.w3.org/2011/webtviptv'>

</webtv>
```

or

```
<html xmlns='http://www.w3.org/2011/webtv'
       xmlns:dvb='http://www.w3.org/2011/webtvdvb'
       xmlns:iptv='http://www.w3.org/2011/webtviptv'>
<body>
  <webtv>
    <dvb:channel id='chann1'>
      <name>ABC</name>
      <dvb:source>onid,pid,sid</dvb:source>
      <dvb:ait>
        <application>http://www.appToLaunch.com
        </application>
        <state>signalled</state>
      </dvb:ait>
    </dvb:channel>

    <iptv:virtualChannel id='chann2'>
      <name>ZDF</name>
      <iptv:channel id='chann2hd'>
        <iptv:source>http://www.zdf.com/streamingSD/
video.m3u</iptv:source>
      </iptv:channel>

      <iptv:channel id='chann2sd'>
        <iptv:source>http://www.zdf.com/streamingSD/
video.m3u</iptv:source>
      </iptv:channel>
    </iptv:virtualChannel>

  </webtv>

</body>
</html>
```

The WebTV object would obviously be created and updated by the middleware thanks to a tight integration (in the same way vice recognitionis handled within Webkit).

- json integration of EPG information : appended to objects (channel)
- you can even imagine plugin to bind native or proprietary content to channels (MHP, ...)
- ease of control & interaction

Pages or resource obviously would be generated or updated on the fly by the system which therefore requires native API integration with e.g. tuner middleware functions.

```
<html xmlns='http://www.w3.org/2011/webtv'
      xmlns:dvb='http://www.w3.org/2011/webtvdvb'
      xmlns:iptv='http://www.w3.org/2011/webtviptv'>
<body>

<div class="video">
  <video><source="#WebTVObject" type="video/x-webtv">
  <div class="panel"></div>
</video>
</div>

</body>
</html>
```

Or in the case of a known Virtual Channel we could use:

```
<video><source="#WebTVObject" type="video/x-webtv">
  <track kind="broadcast" src="chann1HD">
  <track kind="broadcast" src="chann1SD">
<div class="panel"></div>
</video>
```

In addition to existing HTML video states, we will need to add states and attributes related to broadcast specific constraints. Attributes with an asterisk are new

Attribute	Description
*channelTuning	onid, pid, sid triplet or equivalent
*aitTable	AIT Table information (DVB specific)
*EPGInformationObject	In case an EPG data has been linked with the dvb:channel object, or with an higher level, it will also be made available to the panel class for display & processing purpose
*eitObject	Object including informations retrieved from the eit (DVB specific)
readyState	yes or no
seeking	N/A if channel is viewed live (it could be used in PVR mode, still to be determined)
currentTime	
startTime	N/A (PVR)
duration	

Attribute	Description
paused	N/A (PVR)
defaultPlaybackRate	N/A (PVR)
playbackRate	N/A (PVR)
TimeRange played	
TimeRange seekable	
ended	N/A
autoplay	true
loop	N/A
play	N/A (PVR)
pause	N/A (PVR)
controls	enhanced panel with specific channel managements controls (program +/-, ...)
volume	
muted	