CELF Audio Video Graphics Specification v2

San Jose, USA, Tuesday April, 11th 2006

Ruud Derwig – AVGWG chair
Outline

• Introduction
• Requirements
• Technologies and solutions
• Specification
  – Audio
  – Video
  – Graphics
• Status & Next steps
• Summary
• Acknowledgements & References
Specification Objectives

• Reduce fragmentation, provide direction
• Optimized for CE devices
• Support a wide range of products
• Respect CE value chains and business models, enable differentiation and innovation

LinuxWorld: Motorola reports on its experiences with Linux phones
Wednesday April 05, 2006 (06:13 PM GMT)

... Motorola believes that for embedded Linux to grow in mobile phone devices, it must evolve as a platform that multiple handset manufacturers, application vendors, and phone service carriers all build together.

This level of standardization is necessary to foster innovation and accelerate development, he says. ... unifying the lower levels of the platform will help them get their phones to market, but still leave room for them to differentiate their products from one another.

(http://www.mobile.newsforge.com/article.pl?sid=06/04/05/1815218)
Process

• Specification prepared by the CELF Audio Video Graphics Working Group
  – Requirements
  – Technologies & solutions
  – Specification writing, review and rework cycles
  – Working group approval

• Published for review
  – CELF members
  – Public Wiki

• CELF Architecture group review and voting
• CELF Board of Directors voting
Requirements - 1/2

• Global
  – Support wide range of products: Home, Mobile, and Automotive
  – Low-level, platform API
  – Abstract from implementation, enable middleware portability
  – Scalable, deal with diversity of the CE domain
  – Well documented, consistent syntax, semantics, behavior
  – Enable evolution, no unnecessary coupling between interfaces
  – Open API, available without cost, no implementation restrictions

• Linux related
  – Fit with Linux and existing solutions from AVG specification v1.0
  – Allow for efficient implementation on Linux
Requirements - 2/2

• Domain scope related
  – CELF Mobile Phone Profile working group
    • Limited input for this version
  – CELF Digital TV/STB Profile working group
    • Focus of this version
  – Analog and digital broadcast (NTSC, PAL, ATSC, DVB, ISDB, ...)
  – Personal Video Recorders (PVR), Digital Media Adaptors (DMA)
  – Decoding and rendering of formats like wav, midi, jpeg, mpeg, ...
  – Multiple graphics/video planes, alpha blending, ...
  – Simple data services and interactivity (closed caption, teletext, MHEG, MHP, OCAP)
Outline

• Introduction
• Requirements
  • Technologies and solutions
    • Specification
      – Audio
      – Video
      – Graphics
    • Status & Next steps
  • Summary
  • Acknowledgements & References
Universal Home API

- Defined by the UHAPI Forum
- Simplifying software development for CE products
- Application portability over multiple HW/SW platforms
- Open Source implementation: UHAPI4Linux
## UHAPI 1.1 contents

<table>
<thead>
<tr>
<th>General documents (11)</th>
<th>Decoders/Encoders (7)</th>
<th>Audio Processing Components (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Specification Reader’s Guide</td>
<td>ATSC Decoder</td>
<td>Audio Automatic Volume Leveling</td>
</tr>
<tr>
<td>API Naming Conventions</td>
<td>Image Decoding</td>
<td>Audio Bass Enhancements</td>
</tr>
<tr>
<td>Error Handling</td>
<td>SPDIF Decoding</td>
<td>Audio Dynamic Range Control</td>
</tr>
<tr>
<td>Execution Architecture</td>
<td>STC Decoding</td>
<td>Audio Mixing</td>
</tr>
<tr>
<td>Notification</td>
<td>Transport Stream Demultiplexing</td>
<td>Audio Noise Generation</td>
</tr>
<tr>
<td>Qualifiers Quick Reference</td>
<td>Transport Stream Multiplexing</td>
<td>Audio Program Selection</td>
</tr>
<tr>
<td>API Evolution Rules</td>
<td>SPTS Transmuxing</td>
<td>Audio Volume Control</td>
</tr>
<tr>
<td>Get / Set Specification Patterns</td>
<td>Handling variation</td>
<td>Equalizing</td>
</tr>
<tr>
<td>Handling variation</td>
<td>Accessing UHAPI’s functionality and third-party binding</td>
<td>Speaker Set / Headphones</td>
</tr>
<tr>
<td>Accessing UHAPI’s functionality and third-party binding</td>
<td>Handling Disturbed Signals</td>
<td>Output Configuration</td>
</tr>
<tr>
<td>Handling Disturbed Signals</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video Processing Components (15)</th>
<th>Audio Processing Components (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Level</td>
<td>Audio Automatic Volume Leveling</td>
</tr>
<tr>
<td>Analog Video Decoding</td>
<td>Audio Bass Enhancements</td>
</tr>
<tr>
<td>Analog Video Encoding</td>
<td>Audio Dynamic Range Control</td>
</tr>
<tr>
<td>Analog Video Encryption</td>
<td>Audio Mixing</td>
</tr>
<tr>
<td>Basic Video Featuring</td>
<td>Audio Noise Generation</td>
</tr>
<tr>
<td>Black Bar Detection</td>
<td>Audio Program Selection</td>
</tr>
<tr>
<td>Color Transient Improvement</td>
<td>Audio Volume Control</td>
</tr>
<tr>
<td>Dynamic Noise Reduction</td>
<td>Equalizing</td>
</tr>
<tr>
<td>Histogram Modification</td>
<td>Speaker Set / Headphones</td>
</tr>
<tr>
<td>Noise Measurement</td>
<td>Output Configuration</td>
</tr>
<tr>
<td>Scan Rate Conversion</td>
<td></td>
</tr>
<tr>
<td>Sharpness Enhancement</td>
<td></td>
</tr>
<tr>
<td>Sharpness Measurement</td>
<td></td>
</tr>
<tr>
<td>Video Color Enhancement</td>
<td></td>
</tr>
<tr>
<td>Video Mixing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generic/other (10)</th>
<th>Audio Processing Components (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Audio &amp; Video Demodulation</td>
<td>Audio Automatic Volume Leveling</td>
</tr>
<tr>
<td>Analog AV Input</td>
<td>Audio Bass Enhancements</td>
</tr>
<tr>
<td>Anti Aging</td>
<td>Audio Dynamic Range Control</td>
</tr>
<tr>
<td>Analog Audio Decoding</td>
<td>Audio Mixing</td>
</tr>
<tr>
<td>Channel Decoding</td>
<td>Audio Noise Generation</td>
</tr>
<tr>
<td>RF Amplification</td>
<td>Audio Program Selection</td>
</tr>
<tr>
<td>Out Of Band Tuning &amp; Demodulation</td>
<td>Audio Volume Control</td>
</tr>
<tr>
<td>Signal Strength</td>
<td>Equalizing</td>
</tr>
<tr>
<td>Tuning</td>
<td>Speaker Set / Headphones</td>
</tr>
<tr>
<td>HdmiIn</td>
<td>Output Configuration</td>
</tr>
<tr>
<td>SPDIF-in</td>
<td></td>
</tr>
<tr>
<td>VBI Slicing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type specifications (2) :</th>
<th>Audio Processing Components (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Types</td>
<td>Audio Automatic Volume Leveling</td>
</tr>
<tr>
<td>Global Types</td>
<td>Audio Bass Enhancements</td>
</tr>
</tbody>
</table>

### General documents:
- API Specification Reader’s Guide
- API Naming Conventions
- Error Handling
- Execution Architecture
- Notification
- Qualifiers Quick Reference
- API Evolution Rules
- Get / Set Specification Patterns
- Handling variation
- Accessing UHAPI’s functionality and third-party binding
- Handling Disturbed Signals

### API specifications:
- **Front End Components (12)**
  - Analog Audio & Video Demodulation
  - Analog AV Input
  - Anti Aging
  - Analog Audio Decoding
  - Channel Decoding
  - RF Amplification
  - Out Of Band Tuning & Demodulation
  - Signal Strength
  - Tuning
  - HdmiIn
  - SPDIF-in
  - VBI Slicing

### Decoders/Encoders:
- ATSC Decoder
- Image Decoding
- SPDIF Decoding
- STC Decoding
- Transport Stream Demultiplexing
- Transport Stream Multiplexing
- SPTS Transmuxing

### Video Processing Components:
- Ambient Level
- Analog Video Decoding
- Analog Video Encoding
- Analog Video Encryption
- Basic Video Featuring
- Black Bar Detection
- Color Transient Improvement
- Dynamic Noise Reduction
- Histogram Modification
- Noise Measurement
- Scan Rate Conversion
- Sharpness Enhancement
- Sharpness Measurement
- Video Color Enhancement
- Video Mixing

### Generic/other:
- Analog AV Output
- SPDIF-out
- Data Injecting
- Data Extracting
- Connection Management
- Fatal Error Handling
- I am Alive
- Pin Objects
- Unknown
- URL Source

### Type specifications:
- Basic Types
- Global Types
DirectFB

- Open Source project (directfb.org)
- Powerful graphics library for Linux
- Designed with embedded systems in mind
- Compliant to Multimedia Home Platform (MHP)
- Provides HW abstraction
- Allows HW acceleration where possible
- Recommended by the CELF 1.0 specification, adopted by many members
Features

• Drawing (lines, rectangles, filling etc.)
• Blitting (scaled blended, color keyed etc.)
• Font rendering
• Window management
• Multiple application support
  – resource management
• Handling input events
• Scaling
• Color keying
• Layer mixing
OpenGL ES

- Defined by the Khronos group
- Small-footprint subset of OpenGL
- Powerful, low-level API with full functionality for 3D games
ALSA

• Open Source project (alsa-project.org)
• Advanced Linux Sound Architecture
  – audio and MIDI functionality
• Efficient support for all types of audio interfaces
  – consumer soundcards to professional multichannel audio
• Fully modularized sound drivers
• SMP and thread-safe design
• User space library (alsa-lib) to simplify application programming and provide higher level functionality
Outline

• Introduction
• Requirements
• Technologies and solutions
• Specification
  – Audio
  – Video
  – Graphics
• Status & Next steps
• Summary
• Acknowledgements & References
Compliance Classifiers

- Conventions from IETF RFC 2119
  - [M]ust, Required, Shall
  - [S]hould, Recommended
  - [O]ptional, May
  - E[X]pressly Forbidden

- AVG Specification v2.0
  - In good Linux/Open Source tradition: toolbox approach
  - Providing direction by recommendation[S], not [M]andating
Audio – 1/2

- Rational
  - UHAPI has broadest CE domain coverage
  - ALSA(OSS) has best coverage of “PC-domain” (midi, professional)
  - Scalable, extendible, well documented
  - Functional interfaces, implementation independent

- Specification
  [S] Broadcast audio decoding functions: UHAPI
    - Analog Audio Decoding, SPDIF-in, ATSC Decoder, SPDIF-out, SPDIF Decoding
  [S] Audio processing and rendering functions: UHAPI
    - Automatic Volume Levelling, Program Selection, Bass Enhancement, Volume Control, Dynamic Range Control, Equalizing, Mixing, Output Configuration, Noise Generation, Speaker /Headphones
  [S] PC (sound cards) domain functionality: ALSA
    - Synthesizing, MIDI interfaces
  [O] Other domains: existing ALSA and UHAPI interfaces may be used
  [O] Implementations of the UHAPI audio interfaces may internally use ALSA or OSS or other – dedicated – implementations.
Video – 1/3

• Rational
  – UHAPI has broadest coverage for analog and digital broadcast, and PVR
  – UHAPI interface matches well with existing Linux solutions (V4L, LinuxDVB)
  – Video & graphics Mixing: Combine DirectFB and UHAPI
  – Scalable, extendible, well documented
  – Functional interfaces, implementation independent

• Specification
  [S] Broadcast video decoding functions: UHAPI Forum
  • Analog Audio & Video Demodulation, RF Amplification, Analog Video Decoding, Signal Strength, ATSC Decoder, STC Decoder, Analog AV Input, Transport Stream Demultiplexing, Analog AV Output, Transport Stream Multiplexing, Channel Decoding, Tuning, HdiIn, URL Source, Image Decoding 2, VBI Slicing, Out Of Band Tuning & Demodulation
Video – 2/3

• Specification

[S] Video processing and rendering functions: UHAPI
  • Ambient Level, Histogram Modification, Analog Video Encoder, Noise Measurement, Analog Video, Encryption, Scan Rate Conversion 2, Anti Aging, Sharpness Enhancement, Black Bar Detection, Sharpness Measurement, Video Color Enhancement, Basic Video Featuring, Color Transient Improvement, Video Mixing, Dynamic Noise Reduction

[S] PVR video functions: UHAPI Forum
  • Data Injecting, SPTS Transmuxing, Data Extracting

[O] Other domains: Existing UHAPI interfaces may be used

[O] Implementations of the UHAPI interfaces may use existing V4L(2) or LinuxDVB implementations
Video – 3/3

Middleware reads SPTS from HDD and injects it.

uhITuning

uhIChanDec

uhIatscDec

Audio

Digital

uhITuning

uhIAnaAvDem

uhIAnaVdec

uhIAnaAdec

Video

Analog

uhIVmix

Audio

SPTS

Video Mixer

Transmux

Data extracting

Data injecting

ATSC Decoder

SPTS

Video Mixer
Graphics - 1/2

• Rational
  – Widespread use of both DirectFB and standard FrameBuffer
  – UHAPI Video Mixing has most features for composition and scaling
  – OpenGL ES is the de-facto industry standard in the mobile domain and applicable in the home domain
  – Scalable, extendible, well documented
  – Functional interfaces, implementation independent

• Specification
  – [S] The standard Framebuffer is recommended for use in CE devices
  – [S] DirectFB is recommended for use in CE devices
    • [M] IDirectFB, IDirectFBSurface, IDirectFBFont, IDirectFBDisplayLayer, Window, Palette
    • [O] IDirectFBImageProvider, IDirectFBVideoProvider, IDirectFBDataBuffer
    • [X] IDirectFBScreen
  – [S] Graphics combined with video: DirectFB (graphics) combined with UHAPI (video mixing)
  – [S] 3D graphics: OpenGL ES
Graphics – 2/2

- Non-linear scaling
- Render Subtitles
- Color key ranges
- Borders
- Fade to
- ...

CELF 2.0 Application

- Accelerated gfx
- Windowing
- Input devices
- Fonts
- ...

DirectFB implementation

OpenGL ES implementation

Gfx Acceleration Hardware

Vmix

Gfx Layer

DirectFB

OpenGL ES

Audio Video Acceleration Hardware

• Accelerated gfx
• Windowing
• Input devices
• Fonts
• …

UHAPI

• Non-linear scaling
• Render Subtitles
• Color key ranges
• Borders
• Fade to
• …

April 11th, 2006
CELF Embedded Linux Conference 2006
22
Outline

• Introduction
• Requirements
• Technologies and solutions
• Specification
  – Audio
  – Video
  – Graphics

• Status & Next steps
• Summary
• Acknowledgements & References
Status

• CELF AVG working group has finished and approved the specification
• Thursday, April 13th CELF Architecture Group voting
• Formal Board of Directors approval: tbd.

• Open Source implementations available
  – DirectFB
  – ALSA
  – UHAPI interfaces
  – No fully integrated version of DirectFB, UHAPI, ALSA, OpenGL ES
• “Commercial” (partial) implementations available, too
  – Offers from various (semiconductor) vendors
What’s Next

• Stimulate (Open Source) full implementations
  – OpenGL ES and DirectFB
  – UHAPI4Linux on standard PC
    • Add DirectFB
    • … and ALSA, OpenGL ES
  – Members on specific boards

• Extend Specification
  – Media Processing Framework
    • Gstreamer, OpenMAX IL
    • …
  – Mobile domain
    • CELF Mobile Phone Profile WG
    • OpenMAX
    • …
Specification Summary

**Public**, recommended

**Private**, optional implementation

---

**ALSA + UHAPI + DirectFB + OpenGL ES**

... OSS, V4L(2), LinuxDVB, Framebuffer, ...

---

**Audio**

**Video + graphics**
Thank You!

- AVG working group members, especially
  - Aplix
  - “DirectFB.org”
  - Intel
  - Mitsubishi
  - Motorola
  - Nexwave Solutions

- References
  - CELF AVG Specification v2.0: http://tree.celinuxforum.org/pubwiki/moin.cgi/AudioVideoGraphicsSpec_5fV2
  - UHAPI: www.uhapi.org
    - UHAPI v1.1 version, donated by the UHAPI Forum: http://tree.celinuxforum.org/pubwiki/moin.cgi/UHAPI
    - UHAPI4Linux open source implementation: http://sourceforge.net/projects/uhapi4linux/
  - DirectFB: www.directfb.org
    - DirectFB and UHAPI combination: http://tree.celinuxforum.org/CelfPubWiki/AvgUhapiApiTaskForce
  - OpenGL ES: http://www.khronos.org/opengles/spec/
  - ALSA: http://www.alsa-project.org
    - Comparison of Linux Audio interfaces: http://tree.celinuxforum.org/CelfPubWiki/AvgAudioAPIs