Ardour3 Video Integration
film-soundtracks on GNU/Linux

Robin Gareus

CiTu - Pargraphe Research Group
University Paris 8 - Hypermedia Department
robin@gareus.org

April, 2012
Outline of the talk

Introduction

Problem Analysis

API

Video Server

Client Implementation

Coda
Introduction

Soundtrack composition, arrangement and production is rather young discipline.
Introduction

- Soundtrack composition, arrangement and production is rather young discipline
- about 70 years old.
Introduction

Soundtrack composition, arrangement and production is rather young discipline.

About 70 years old.

Rapidly changing technology.
Introduction

- Soundtrack composition, arrangement and production is rather young discipline
- Nevertheless, standard procedures have evolved
Introduction

- Soundtrack composition, arrangement and production is rather young discipline
- Nevertheless, standard procedures have evolved
- Audio and Video remain separated for historically and practically reasons.
Introduction

- Soundtrack composition, arrangement and production is rather young discipline
- Nevertheless, standard procedures have evolved
- Audio and Video remain separated for historically and practically reasons.

- different qualifications and expertise is needed
Soundtrack composition, arrangement and production is rather young discipline.

Nevertheless, standard procedures have evolved.

Audio and Video remain separated for historically and practically reasons.

different qualifications and expertise is needed.

too much gear on the camera.
Introduction

Soundtrack composition, arrangement and production is rather young discipline.

Nevertheless, standard procedures have evolved.

Audio and Video remain separated for historically and practically reasons.

different qualifications and expertise is needed

too much gear on the camera

job-definition by unions
Soundtrack composition, arrangement and production is rather young discipline. Nevertheless, standard procedures have evolved. Audio and Video remain separated for historically and practically reasons. Different qualifications and expertise is needed. Too much gear on the camera. Job-definition by unions. More choices WRT to equipment.
Dialogue (on-camera voice), field-recordings (usually recorded on set - but: overdubs, translations)
Film-Sound

- Dialogue (on-camera voice), field-recordings (usually recorded on set - but: overdubs, translations)
- sound-effects (Foley, Sound-scapes, Spot-sounds - tight sync)
Film-Sound

- Dialogue (on-camera voice), field-recordings (usually recorded on set - but: overdubs, translations)
- sound-effects (Foley, Sound-scapes, Spot-sounds - tight sync)
- film-music
Dialogue (on-camera voice), field-recordings (usually recorded on set – but: overdubs, translations)

sound-effects (Foley, Sound-scapes, Spot-sounds - tight sync)

film-music

One goal is to synchronize dramatic events happening on screen with musical events in the score.
Film-Sound

- Dialogue (on-camera voice), field-recordings (usually recorded on set - but: overdubs, translations)
- sound-effects (Foley, Sound-scapes, Spot-sounds - tight sync)
- film-music

One goal is to synchronize dramatic events happening on screen with musical events in the score. With only a few exceptions - namely song or dance scenes - music composition and sound-design usually takes place after recording and editing the video.
Post-Production Work-flow

- synchronize the original audio recorded on the set with the edited video
Post-Production Work-flow

- synchronize the original audio recorded on the set with the edited video
- design, arrange and align sound-effects
Post-Production Work-flow

- synchronize the original audio recorded on the set with the edited video
- design, arrange and align sound-effects
- compose, record and edit music
Post-Production Work-flow

- synchronize the original audio recorded on the set with the edited video
- design, arrange and align sound-effects
- compose, record and edit music
- mix-down and master the soundtrack (various versions: TV-compressed, 5.1 cinema,..)
A/V sync

- analog: audio-visual cues (slate)
A/V sync

- analog: audio-visual cues (slate)
- digital: time-code (most commonly SMPTE produced by the camera)
A/V sync

- analog: audio-visual cues (slate)
- digital: time-code (most commonly SMPTE produced by the camera)
- digital: EDL, AAF, MXF, BWF, OMF+OMFI,...
A/V sync

- analog: audio-visual cues (slate)
- digital: time-code (most commonly SMPTE produced by the camera).
- digital: EDL, AAF, MXF, BWF, OMF+OMFI,...

Technical skills and details involved can become quite complex.
A/V sync

- analog: audio-visual cues (slate)
- digital: time-code (most commonly SMPTE produced by the camera).
- digital: EDL, AAF, MXF, BWF, OMF+OMFI,...

Technical skills and details involved can become quite complex. Neither composers nor sound-designers do want to concern themselves with that task.
Tools

- Digital Audio Workstation
Tools

- Digital Audio Workstation
- Synchronous video playback
- Video-timeline
Tools

- Digital Audio Workstation
- Synchronous video playback
- Video-timeline
- Session-management
Tools

- Digital Audio Workstation
- Synchronous video playback
- Video-timeline
- Session-management
- Import A/V (timecode, edl, audio-extract)
- A/V alignment (offset, pull-up/down, timecode-conversion)
- Export A/V
Goals

- easy-to-use
Goals

- easy-to-use
- professional
Goals

- easy-to-use
- professional
- workflow for film-sound production
Goals

- easy-to-use
- professional
- workflow for film-sound production
- free-software.
Goals

- easy-to-use
- professional
- workflow for film-sound production
- free-software.

More specifically: Integration of video-elements into the Ardour Digital Audio Workstation.
Goals

- easy-to-use
- professional
- workflow for film-sound production
- free-software.

More specifically: Integration of video-elements into the Ardour Digital Audio Workstation. The resulting interface must not be limited to the software at hand (Ardour, Xjadeo, icsd) but allow for further adaption or interoperability.
Design

- Client-Server model
  - external video monitoring app.
  - external video-decoder application
Design

- Client-Server model
  - external video monitoring app.
  - external video-decoder application
- “Video-server”
  - video-frame-cache
  - video session management
  - modular design - pluggable video decoder back-end
Design

Figure: Design overview
API Overview

All requests from GUI to backend are done via HTTP
API Overview

All requests from GUI to backend are done via HTTP

- HTTP is a well supported protocol with existing infrastructure
- established proxy and load-balancing systems
- persistent HTTP connections
- web-interface
- ..but: no out-of-band communication
API Overview

All requests from GUI to backend are done via HTTP

- HTTP is a well supported protocol with existing infrastructure
- established proxy and load-balancing systems
- persistent HTTP connections
- web-interface

..but: no out-of-band communication

Requests handlers:

- info (file and/or session information)
- image (video-frame, thumbnail)
API Overview

All requests from GUI to backend are done via HTTP

- HTTP is a well supported protocol with existing infrastructure
- established proxy and load-balancing systems
- persistent HTTP connections
- web-interface
- ..but: no out-of-band communication

Requests handlers:

- info (file and/or session information)
- image (video-frame, thumbnail)
- stream (export, render)
- admin (cache-flush, status,...)
API parameter units

- **Time:** video-frame count (duration, offset, time)
- **Geometry:** effective image size (incl. pixel-aspect and display-aspect)
- **Framerate:** ratio
- **Image:** various-formats (raw RGB[A], PNG, JPG, YUV..)
- **Text:** serialized key-value store in various formats (XML, JSON,..)
API parameter units

- **Time**: video-frame count (duration, offset, time)
- **Geometry**: effective image size (incl. pixel-aspect and display-aspect)
- **Framerate**: ratio
- **Image**: various-formats (raw RGB[A], PNG, JPG, YUV..)
- **Text**: serialized key-value store in various formats (XML, JSON,..)

The reply-format is chosen by the file-extension and may be overridden using request parameters.
Info Request

Requesting information about a session of file:

- file-name or session-name (required)
- reply-format (optional, implicit)
Video-Frame Request

Requesting a single video-frame

- file-name or session-name (required)
- frame the frame-number (starting at zero for the first frame - required).
- width, height (optional)
- reply-format (optional, implicit)
Server Implementation

- HTTP server in POSIX-C
- a video-frame cache
- multiple decoder instances:
  - parallel decoding
  - efficiency - keep state, key-frame continuity
- tested on GNU/Linux, OSX and win32
Server Performance

- Cache hits are dominated by transfer time (few ms - depending on image-size and network)
Server Performance

- Cache hits are dominated by transfer time (few ms - depending on image-size and network)
- Decoder latency is highly dependent on the geometry of the movie, codec, CPU and I/O.
Cache hits are dominated by transfer time (few ms - depending on image-size and network)

Decoder latency is highly dependent on the geometry of the movie, codec, CPU and I/O.

On slower CPUs ($\leq 1.6$ GHz Intel) a full HD Video can be decoded and scaled at 25 fps using mjpegs codec width only intra-frames at the cost of high I/O.
Server Performance

- Cache hits are dominated by transfer time (few ms - depending on image-size and network)
- Decoder latency is highly dependent on the geometry of the movie, codec, CPU and I/O.
- On slower CPUs ($\leq 1.6$ GHz Intel) a full HD video can be decoded and scaled at 25 fps using mjpeg codec width only intra-frames at the cost of high I/O.
- Faster CPUs can shift the load towards the CPU.
Server Performance

- Cache hits are dominated by transfer time (few ms - depending on image-size and network).
- Decoder latency is highly dependent on the geometry of the movie, codec, CPU and I/O.
- On slower CPUs (≤ 1.6 GHz Intel) a full HD video can be decoded and scaled at 25 fps using mjpeg codec width only intra-frames at the cost of high I/O.
- Faster CPUs can shift the load towards the CPU.
- Parallelizing requests increases latency to up to a few hundred milliseconds.
Cache hits are dominated by transfer time (few ms - depending on image-size and network)

Decoder latency is highly dependent on the geometry of the movie, codec, CPU and I/O.

On slower CPUs ($\leq 1.6$ GHz Intel) a full HD video can be decoded and scaled at 25 fps using mjpeg codec width only intra-frames at the cost of high I/O.

Faster CPUs can shift the load towards the CPU.

Parallelizing requests increases latency to up to a few hundred milliseconds.

This is intended behavior: image for a whole view-point page will arrive simultaneously.
Server Performance

**Figure:** latency for decoding 80x60 thumbnails for 1, 2 and 8 parallel requests.

**Figure:** decoder and request latency for PAL (768x576px) video.

Robin Gareus (CiTu) Ardour3 Video Integration April 2012 17 / 21
Changes to Ardour3:

- Video-timeline display (unique “ruler”)

Client Implementation
Client Implementation

Changes to Ardour3:

- Video-timeline display (unique “ruler”)
- HTTP interaction with the video-server
Client Implementation

Changes to Ardour3:

- Video-timeline display (unique “ruler”)
- HTTP interaction with the video-server
- Xjadeo remote control (video-monitor)
Client Implementation

Changes to Ardour3:

- Video-timeline display (unique “ruler”)
- HTTP interaction with the video-server
- Xjadeo remote control (video-monitor)
- ffmpeg CLI interaction (import/export)
Client Implementation

Changes to Ardour3:

- Video-timeline display (unique “ruler”)
- HTTP interaction with the video-server
- Xjadeo remote control (video-monitor)
- ffmpeg CLI interaction (import/export)
- Dialogs (the largest part)
Changes to Ardour3:

- Video-timeline display (unique “ruler”)
- HTTP interaction with the video-server
- Xjadeo remote control (video-monitor)
- ffmpeg CLI interaction (import/export)
- Dialogs (the largest part)
- Support and helper functions
Outlook

- the building-blocks are unit-tested and some real-world feedback from a handful of devs/artists/engineers.
Outlook

- the building-blocks are unit-tested and some real-world feedback from a handful of devs/artists/engineers.
- take it to end-users (post Ardour3.0 release): “as soon as 3.0 is released, we’ll merge his work into the mainstream code.” (Paul D.)
Outlook

- the building-blocks are unit-tested and some real-world feedback from a handful of devs/artists/engineers.
- take it to end-users (post Ardour3.0 release): “as soon as 3.0 is released, we’ll merge his work into the mainstream code.” (Paul D.)
- video-server: (GNU/Linux) all dependencies are already in Debian/stable - packaging the video-server is prepared - pending last minute changes
● the building-blocks are unit-tested and some real-world feedback from a handful of devs/artists/engineers.

● take it to end-users (post Ardour3.0 release): “as soon as 3.0 is released, we’ll merge his work into the mainstream code.” (Paul D.)

● video-server: (GNU/Linux) all dependencies are already in debian/stable - packaging the video-server is prepared - pending last minute changes

● video-server: statically linked version for win32 and OSX available.
ToDo

- update to latest A3 internal API (in particular: audio export)
- multi-track export and mux (5.1)
- video-file “import” (hard-link, session-folder, keep orig and transcoded)
- expose EDL support in UI; ardour: AAF, MXF, BWF, EDL...
- messages (fix Inglisch :-) and translate)
- user-definable video-export presets
- optimizations (thread-pool for image requests, ..., HTTP Keep-Alive)
- squash bugs: A3: 8k LOC ; icsd: 12k LOC