Integrating videoconferencing into everyday applications

Olivier Crête

Collabora
Ingredients

- Call signallings
  - Telepathy
- Media Framework
  - GStreamer
- Call protocol abstraction
  - Farsight 2
Telepathy: Overview

Chat Client
Contact List
VOIP Client

XMPP Backend
SIP Backend
Foobar IM Backend
Telepathy: Concepts

Connection Manager

Connection

IM Channel

Media Channel

Contact List Channel
Introducing GStreamer

- Assemble media pipelines from elements
- Four major types of elements:
  - Sources (camera, microphone, file, etc)
  - Sinks (screen, speakers, file, etc)
  - Transformational elements (encoder, decoders, multiplexers, demultiplexers, etc)
  - Also, complex “magical” elements (playbin, decodebin, Farsight2, etc)
Basic GStreamer

- Elements

  - filesrc
  - mad
  - alsasink
Basic GStreamer

- Elements
- Bins

```
bin
  filesrc
  mad
  alsasink
```

[Image of GStreamer elements and bins diagram]
Basic GStreamer

- Elements
- Bins
- Pads
More Complex Pipelines

Pipeline

1. filesrc
2. src
3. sink
4. oggdemux
   - src1
   - src2
5. decodebin
6. theoradec
   - src
   - sink
7. vorbisdec
   - src
   - sink
8. alsasink
9. xvimagesink
   - sink
Farsight 2: Overview

- Audio & video conferencing abstraction
- Multiple protocols abstracted
  - RTP (XMPP Jingle, SIP, etc)
  - MSN Webcam (almost there)
  - Others...
- GStreamer element
High level objects

- Participants
  - People
- Sessions
  - Media types (audio, video)
- Stream
  - Intersection of Participant and Session
Farsight 2: RTP

- Complete RTP with RTCP
  - Including A/V sync
- DTMF, codec negotiation,
- Multiple transports for RTP
  - ICE (standard, Google and MSN)
  - Unicast UDP
  - Multicast UDP
  - Application (soon)
Telepathy Farsight integration

- Telepathy-Farsight library
- Hides all signalling
  - Easy API
- Telepathy streamed media interface limited
  - Only one-to-one
  - Strange session/stream separation
Farsight 2 element

My Audio → sink_1 → src_1_3213212_8 → Remote Audio
          → src_1_3213212_96 → Remote Audio
          fs*conference

My Video → sink_2 → src_2_9875203_34 → Remote Video

Integrated Farsight 2: Simple Case

pulsesrc

src

sink_1

sink

fs*conference

src_1_3213212_8

sink

pulsesink

src_1_3213212_96

sink

sink

src_2_9875203_34

sink

xvimagesink

src

sink_2

src_2_9875203_34
Another simple case

alsasrc

fs*conference

liveadder

alsasink

v4l2src

xvimagesink
Add recording

- pulsesrc
- v4l2src
- fs*conference
- tee
- vorbisenc
- theoraenc
- oggmux
- filesink
- xvimagesink
- pulsesink
Steps to get a Telepathy call

1. Get D-Bus bus
2. Get Telepathy Connection & Contact
3. Find the handle for that contact
4. Get a Telepathy Channel
5. Wrap it into Telepathy-Python
6. Wrap it into Telepathy-Farsight
7. Attach session-created and session-created
8. Add members
9. Request streams
Steps to get a Telepathy call

10. Put the conference into the pipeline on Telepathy session creation

11. Add/Link the sink/source on stream creation
Extras

- Hook up UI to add added streaming-side features
- Example:
  - The Record button adds the recording pipeline
What's missing

• Mission Control 5
  • New Telepathy Specification
  • Allows for proper channel dispatching
• New PiTiVi
  • More integration
    – Input
    – Output
What else?

- Collaborative text editing
  - OOo, Abiword, Gnumeric
- Stream slides with Evince
- Remote desktop viewing
- Other ideas?
Thank you

- #farsight, #telepathy, #gstreamer, #pitivi @ FreeNode
- http://farsight.freedesktop.org/
- http://telepathy.freedesktop.org/
- http://gstreamer.net/