GStreamer in OpenOffice.org?

Cédric Bosdonnat,
Radek Doulík

Introduction

Cédric Bosdonnat

- INSA Lyon

Radek Doulík

- Novell, Inc.

Overview

GStreamer library introduction

Macros using GStreamer

Media handling in OpenOffice.org

avmedia with GStreamer backend

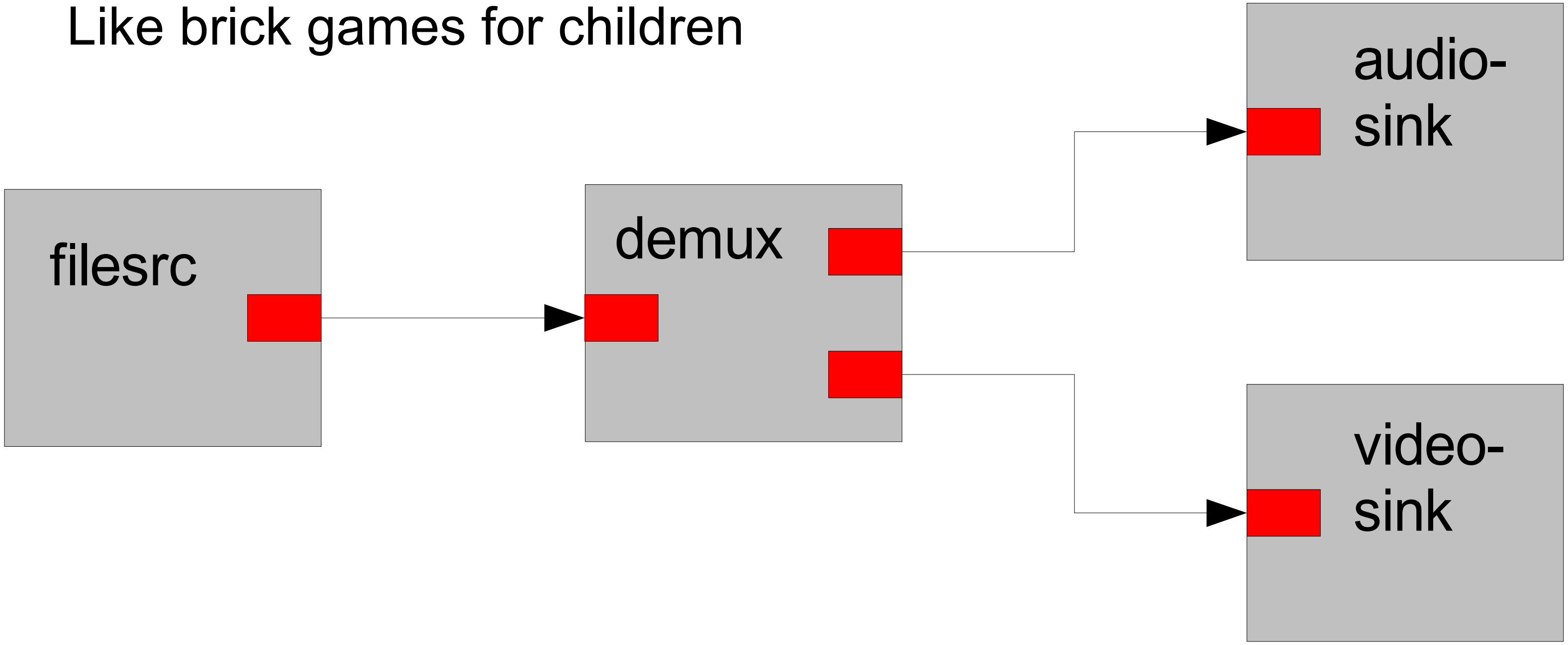
Future work

Questions / Discussion

GStreamer library introduction

Multimedia framework library

- Using pipelines and linked bins to handle the stream
- Many plugins available, even for video editing
- Free software
- Like brick games for children



Macros with GSreamer

Macros with GStreamer

Component Context

- Goal, objectives
- History

Component Design

- An independent component
- A wrapper for the GStreamer API

Component Use

- Build up the pipeline
- Control the pipeline

Component Context

Goal, Objectives

- Running GStreamer pipelines from macros
- Providing a generic pipeline-based multimedia API
- Better control of the multimedia handling

History

- Developing an URE based application at INSA
- Handling multimedia
- Use of GStreamer

Component design

A wrapper for the GStreamer API

- Providing an API to handle multimedia through pipelines
- GStreamer implementation is work in progress
- Could be adapted to other pipeline-based libraries

An independent component

- To be usable in any UNO based application
- Copied some comphelper useful classes and templates

Using the component

Build up the pipeline

- Add the needed bins to the pipeline
- Set their properties
- Create the links between the bins

Control the pipeline

- Simply change the pipeline state to:
  - PLAYING
  - PAUSE
- Send an event on the pipeline bus to control
  - Current position in the stream
  - Stream reading speed

Media handling in OpenOffice.org

Redundancy

- VCL Sound
- avmedia

VCL Sound

- around 5 unix backends
- another 2 shared library loaded on start

avmedia

- UNO based
- audio/video playback

avmedia GStreamer backed

avmedia with GStreamer backend

Plan

- create new backend for avmedia which will use Gstreamer
- UNO magic and Player first, MediaWindow and FrameGrabber later

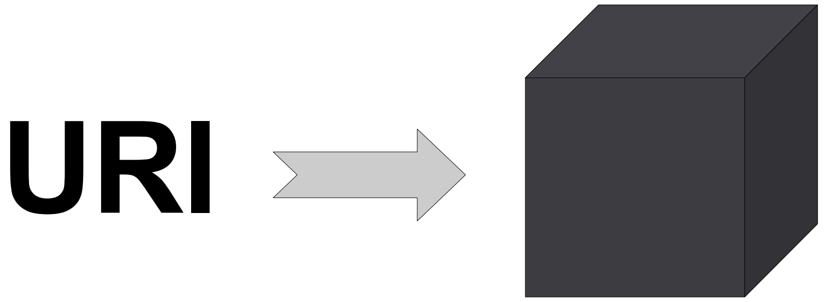
Benefits

- one unix backend
- many supported formats
- GStreamer is easily extensible by plug-ins
- threaded
- well documented
- easy to use

avmedia with GStreamer backend

Inside

- magic black box called PlayBin – GStreamer pipeline
- you feed it with URI and it does everything for you (mostly :-)
- stream auto detection – links all the necessary elements together
- half the code just forwards requests to our black box Playbin
- uri, seeking, volume management, state management



URI

avmedia with GStreamer backend

Inside (continued)

- we watch the pipeline bus and use some of passing messages
- set the X window ID
- state changes

avmedia with GStreamer backend

OOo sync. vs. GStreamer async.

- OOo UNO API works synchronously
- while GStreamer is threaded and works asynchronously

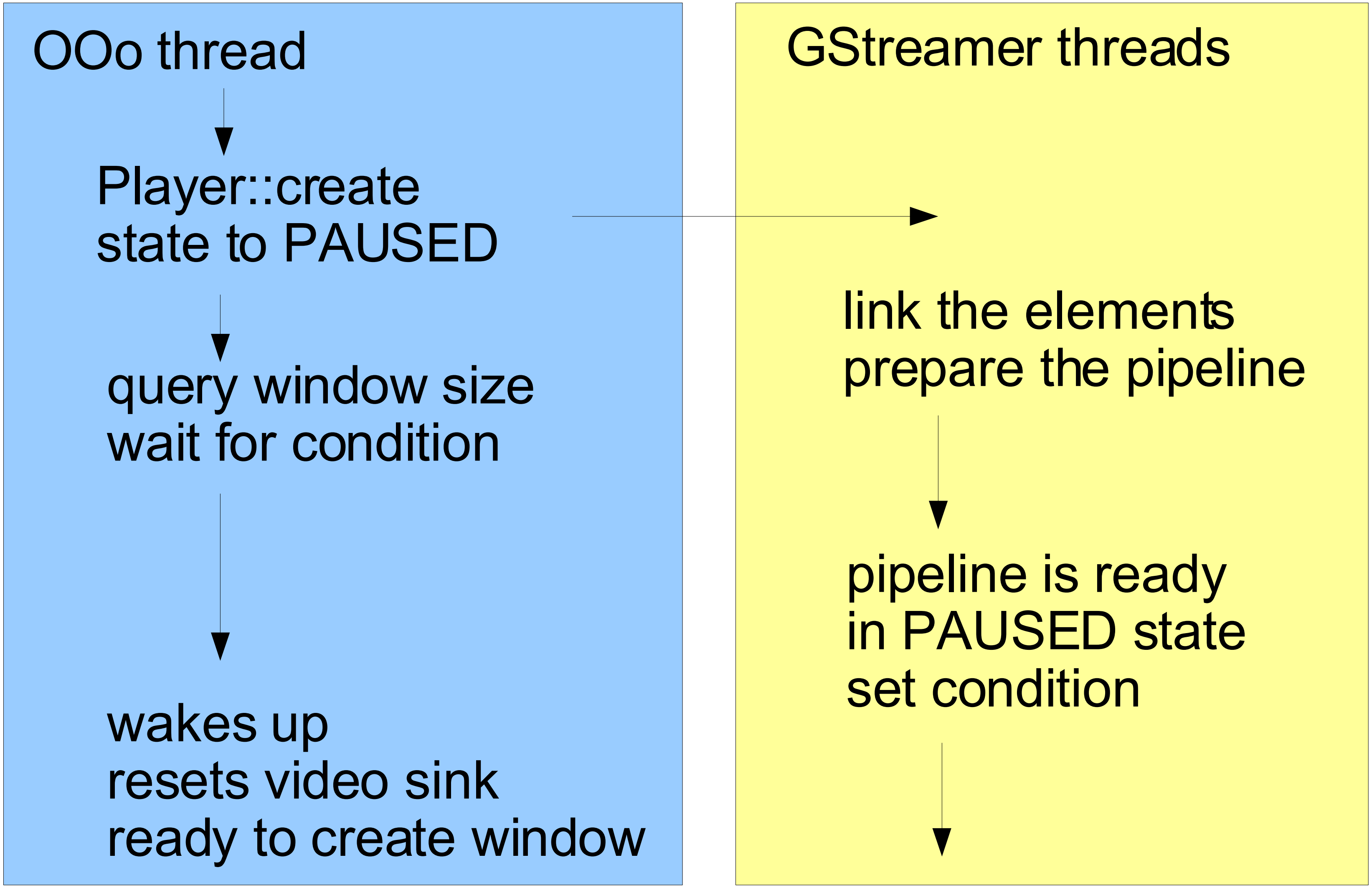
Issues

- we need to provide X window ID to the video sink
- when playback window is created we need to know the video size

Solution

- in the beginning we tell the Playbin to use fake videosink
- wait until we reach PAUSED state

avmedia with GStreamer backend



OOo thread

Player::create
state to PAUSED

query window size
wait for condition

wakes up
resets video sink
ready to create window

GStreamer threads

link the elements
prepare the pipeline

pipeline is ready
in PAUSED state
set condition

Examples

Example 1

Photos slideshow with background music

- current limitation: sound objects cannot live across slide boundaries
- temporary solution: let animate photos on one slide
- audio was edited in audacity (trim, fade in) http://audacity.sourceforge.net/
- sound was recorded in the morning in the forest. photos taken at the same time

The end

Example 2

Include captured video in your presentation

- recorded with xvidcap http://xvidcap.sourceforge.net/
- shows how the previous example was created

Future work

Kill vcl Sound?

- few possibilities
- replace it with avmedia.Player whenever used
- add GStreamer backend to it
- replace it with another simple UNO component

New features

- sound effects across multiple slides (see issuezilla)
- include the media files inside the document
- fade in/fade out effects
- trimming
- suggest one

Questions and discussion

Novell®

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.