GNOME Multimedia Framework

Elliot Lee

sopwith@redhat.com

The Problem

Linux has historically lagged behind other operating systems in the multimedia arena. Microsoft's Windows™ and Apple's MacOS™ both have strong support for multimedia devices, multimedia content creation, playback, and realtime processing. Linux, on the other hand, has a poorly integrated collection of multimedia utilities and applications available, which can hardly compete with the professional level of software available for MS Windows and MacOS.

The Linux user who wishes to hear a sound file must hunt through their collection of sound file players in order to play the tens of sound file formats in wide use today. The Linux developer who wishes to embed a video clip in their application must use crude hacks to run an external video player. The sound designer who wishes to record and edit digital audio in a stable, high-performance environment must dismiss Linux as an option.

Enter GMF

The GNOME Media Framework (hereafter abbreviated to GMF) provides a framework for recording, processing, and presenting media data in a type-independant manner. It is designed to be used for professional audio processing and real-time performance, while at the same time allowing simple applications such as media players. GMF addresses the problem of lackluster Linux media support by building a strong multimedia framework, upon which advanced and integrated multimedia software can be built.

Possible uses of GMF include:

- A media player that can play back a wide variety of media types (sound and video files, streaming network audio/video, DVDs, CD-ROMs, and more) through a single user interface.
- A video editor that allows pluggable special effects filters.
- A music performance in which prerecorded light patterns are synchronized with input from live performers.
- Batch jobs to add multiple effects to a set of sound files.
- A Bonobo component to allow developers to embed multimedia content in their applications.

How it works

GMF implements multimedia-specific objects on top of the GNOME CORBA framework. It utilizes many design ideas from Microsoft's DirectShow™ technology, but is meant to make full use of the features available in the GNOME libraries.

The most basic concepts behind GMF are those of filters and pipes. Filters are objects that produce, process, and/or render media data. There are three types of filters:

- Source filters read media data from an external source, such as a file or a hardware interface.
- Transform filters take input from source filters or other transform filters, and manipulate the data in some way.

- Rendering filters are the last link in the chain, taking on the task of actually displaying the media data, or outputting it to a file or hardware interface.

Pipes are the connections between filters. They take care of transferring the output of one filter to the input of another filter, negotiating the most efficient data transport mechanism possible. Currently, GMF pipes can send data via CORBA calls and UNIX sockets, and much of the infrastructure is in place to support the TCP/IP and shared memory transports. For hardware devices that connect directly to each other, or for application-specific purposes, custom transports are also supported.

Related filters are collected together in the filter graph, which provides a way of controlling and inspecting all the filters, as well as providing a common time reference for synchronization purposes. This synchronization is an important aspect of multimedia applications. The GMF time reference system and the timestamps given to media samples allow applications to ensure synchronized rendering to multiple render filters.

Status and Conclusion

Version 0.1 of the GNOME Media Framework is currently available as a developers-only release. The basic `libgmf` is written and lightly tested, small MIDI filters have been written and tested, a basic filter graph editor is running, and the beginnings of a media player exist.

The major issues remaining are:

- [Design] Proper synchronization with real-time inputs, while still allowing preprocessing.
- [Design] Allowing a generic file or stream data source filter that knows how to reposition the right amount for the media type it is outputting.
- [Code] Saving filter graphs to XML files, and loading them back from XML files.
- [Code] Writing the media player.
- [Code] Writing source filters for all the major hardware devices and types of media files.
- [Code] Writing rendering filters for all the major hardware devices.
- Documentation for programmers wishing to write GMF filters and applications.

With effort, the GNOME Media Framework will not only give Linux applications the same level of multimedia support as Windows and MacOS applications have, but allow the development of programs that were never possible on other operating systems. By paying attention to multimedia support, the ability of the Linux desktop environment to effectively present information to the user is assured.