sipXmediaLib Notes

sipXmediaLib - the media processing library

The sipXmediaLib includes all of the audio processing used in the **sipXtapi**, **sipXezPhone** and **sipXvxml** projects. For example, the library contains audio bridges, audio splitters, echo supression, tone generation (e.g. DTMF), streaming support, RTCP, audio codecs, etc.

The highest level object in the sipXMediaLib is the MpFlowGraph. The flow graph assembles a number of media resources in a defined order of media flow where each resource has zero or more inputs and zero or more outputs. The resouces are process in the oreder implied by the connection topology. The following illustrates the logical connections of the MpCallFlowGraph. Resources may be added dynamically to the flowgraph and new resouce types can be derived. A connection is a logical construct containing the resources related to a single remote media source. There may be zero or more connections in a MpCallFlowGraph.

Internals

(This is basically a capture for posterity of a posting from the email list. It should be polished and improved over time.)

sipX can perform real time audio processing to handle audio. This is done within **sipXmediaLib**. It is here where various manipulations of the audio stream are handled, including compression codecs, echo cancellation, etc.

The basic organization of sipxmedialib is as follows:

There are two threads that get started from dmaTask.cpp that handle interfacing to media input/ouput devices (mic and speaker). One thread pumps data to the speaker, the other pumps data from the mic. Look at SpeakerThreadWnt.cpp and MicThreadWnt.cpp for Windows version and at dmaTaskPosix.cpp for Linux version.

- There's another thread that gets started in NetInTask that listens on RTP sockets for incoming packets, and then hands them to the flow graph (through MprFromNet).
- The signal processing is done through a pipeline called a Flow Graph. Resources, such as codecs, are strung together by adding, and then connecting them into a MpCallFlowGraph object.
- MpConnection creates new MprFromNet and MprToNet objects that pump RTP packets between the net and the call's flowgraph. MprFromNet also coordinates with NetInTask so that it will listen on the appropriate ports.
- The flowgraph is managed by MpMediaTask, and calls doProcessFrame on each resource in the graph in response to a call to MpMediaTask::signalFrameStart(). In Windows this call comes from the speaker thread (SpeakerThreadWnt.cpp) and in Linux from MediaSignaler thread (dmaTaskPosix.cpp).
- Codecs are held in MprDecode and MprEncode objects, so take a look at their doProcessFrame methods for a clue about how they pump data through the codec.

One thing that may be a bit confusing when you first look at this stuff is the way threads and messaging are used. sipX creates a *lot* of threads. Each class that runs in its own thread descends from OsServerTask. Inter-thread communications is handled by having "manipulator" methods that post messages to a thread's queue, where they are then dispatched to an appropriate handler. The manipulators and handlers are in the same class; take a look at the class's handleMessage method (it gets called by the base class when a message is available in the queue) to see how messages are dispatched.

Much of the medialib structure was established due to the use by Pingtel of media processing components from Global IP Sound (GIPS). Because the GIPS code is not open source, it was excised when Pingtel moved SIPx to open source, but the framework that was created to support them remains. It is a reasonbly extensible framework, at the cost of a steep learning curve.

Basic replacements for the GIPS components were provided (e.g. an echo canceller), but these are areas that could benefit from additional work to improve them.

(Thanks to Brian Cuthie at Systemix for the above info.)

About MpFlowGraphBase and MpCallFlowGraph

The MpFlowGraphBase is a skeleton used by the MpMediaTask for generic media processing. Essentially, it consists a network of media resources, which are generic processing blocks of media operations. The MpCallFlowGraph is derived from that base class, and consists of a particular set of media resources that are each derived from the generic resource base class to perform various media processing functions. The MpFlowGraphBase provides the basic manipulations that the MpMediaTask performs without any inherent knowledge of the processing performed by the flowgraph, while the MpCallFlowGraph provides to the application code an interface for manipulations of the audio processing for a call (for example, add or remove a remote connection, start or stop sending RTP, start or stop playing a tone).