

## 5

# Integração da Ferramenta de Ajuste com Exibidores de Conteúdo

Conforme explicado no Capítulo 4, a ferramenta de ajuste processa dados de áudio para futuro armazenamento ou disponibilização do fluxo processado diretamente para uma outra aplicação. Deve existir um mecanismo de comunicação capaz de integrar exibidores de conteúdo com a ferramenta de ajuste, seja utilizando um arquivo gerado ou recebendo o fluxo de bytes produzido. Este Capítulo descreve a implementação realizada com o objetivo de integrar o algoritmo de ajuste elástico a diferentes exibidores de conteúdo. A Figura 66 ilustra as classes principais para esse propósito.

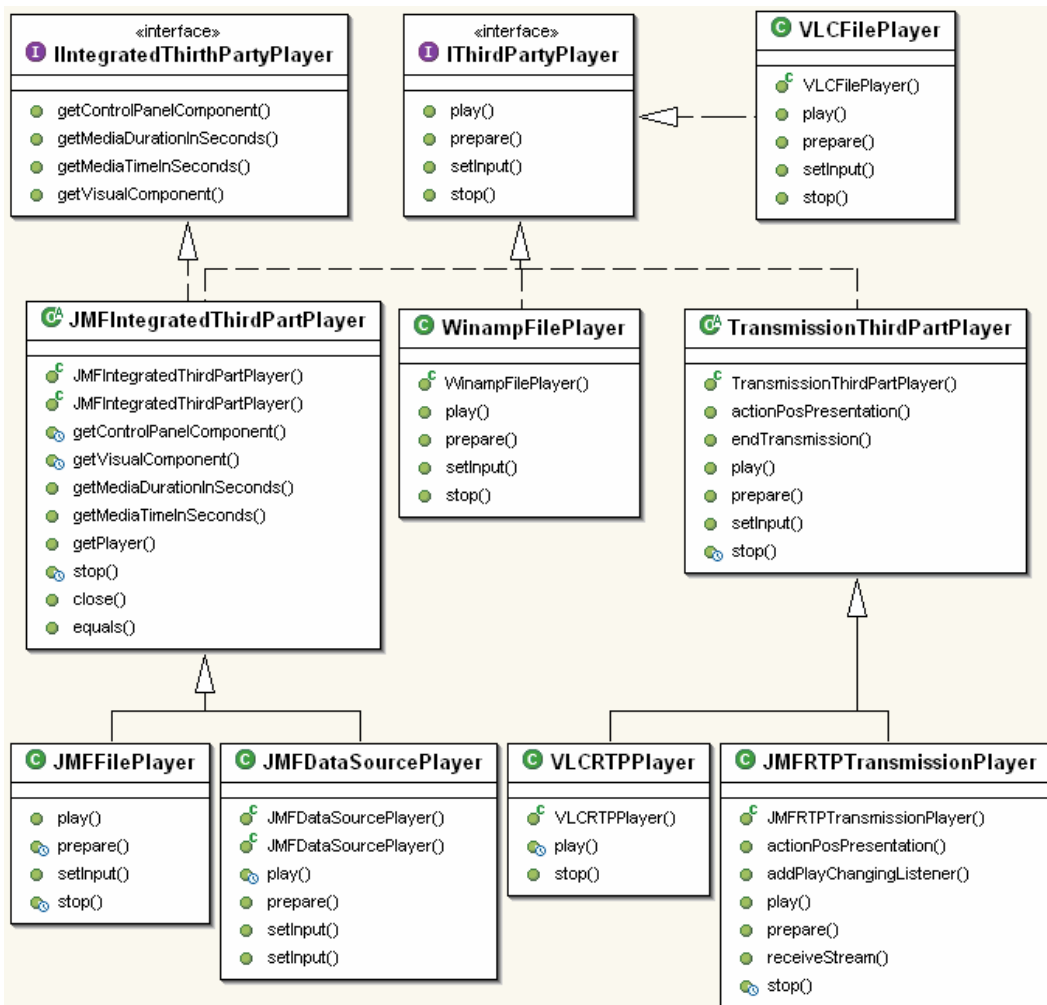


Figura 66 - Diagrama das classes dos exibidores de conteúdo.

Todas as classes que realizam a comunicação da ferramenta de ajuste com exibidor implementam a interface `IThirdPartyPlayer`. As demais classes da Figura 66 serão explicadas nas próximas subseções, que detalham o mecanismo de comunicação utilizado quando ajuste ocorre em tempo de compilação e de execução.

### 5.1. Mecanismo de Comunicação utilizado com Ajuste Elástico em Tempo de Compilação

Quando o ajuste é realizado em tempo de compilação, a integração com o exibidor de conteúdo é realizada utilizando a URI do arquivo contendo a mídia processada. As classes `JMFFilePlayer`, `VLCFilePlayer` e `WinampFilePlayer` foram desenvolvidas para prover a comunicação em tempo de compilação com os exibidores JMF (Sun, 1999), VLC (Videolan, 2005) e Winamp (Nullsoft, 2005), respectivamente.

A classe `JMFFilePlayer` instancia um objeto do exibidor de conteúdo do JMF utilizando a URL do arquivo ajustado. O exibidor do JMF é capaz de reproduzir arquivos nos formatos MP2, MP3, MPEG-1/2 vídeo e sistemas no sistema operacional Windows. Já no sistema operacional Linux, o JMF somente oferece suporte à reprodução de arquivos MP2. As solicitações ao JMF são realizadas via programação Java. Para o JMF fornecer suporte a outros formatos de mídia, é possível acrescentar *plug-ins* capazes de manipular tais formatos. Por exemplo, existe um *plug-in* da empresa SUN (Sun, 1999) que fornece suporte a arquivos MP3 no Linux. Este trabalho utilizou o *plug-in* `Jffmpeg` (Jffmpeg, 2005), que acrescenta ao JMF a capacidade de manipular vários arquivos e fluxos de mídia, como arquivos MP3 e MPEG-1 vídeo e sistemas no Linux. Grande parte das funções do `Jffmpeg` é resultado da tradução de funções do conhecido codificador `Ffmpeg` (Ffmpeg, 2005) (escrito em linguagem C), utilizando JNI (*Java Native Interface*).

O VLC é um exibidor de conteúdo gratuito que oferece suporte a diversos formatos de arquivos, incluindo todos os formatos manipulados por este trabalho. A interação com o VLC para solicitar a exibição de um arquivo foi realizada pela classe `VLCFilePlayer` via um processo externo. O exibidor Winamp é capaz de

reproduzir todos os formatos manipulados por este trabalho, exceto o AC-3. Para interagir com o Winamp, a classe `WinampFilePlayer` realiza uma chamada via um aplicativo chamado `CLamp.exe`<sup>12</sup>, que interpreta um comando escrito e envia a ação correspondente a esse exibidor de conteúdo.

## 5.2. Mecanismo de Comunicação utilizado com Ajuste Elástico em Tempo de Execução

As classes que interagem com exibidores em tempo de execução recebem um fluxo de dados e solicitam imediatamente sua reprodução. Dois modos diferentes para entregar o fluxo ao exibidor de conteúdo foram utilizados: via programação e via RTP.

A interação via programação foi obtida através da classe `JMFDataSourcePlayer`, que envia dados de mídia ajustados para um exibidor de conteúdo do JMF. Nessa interação, algumas classes precisaram ser criadas para definir uma fonte de dados, a ser alimentada pelo algoritmo de ajuste, com a interface que o exibidor de conteúdo espera para buscar a mídia, conforme definido pelo padrão JMF. As classes implementadas para esse fim são ilustradas na Figura 67.

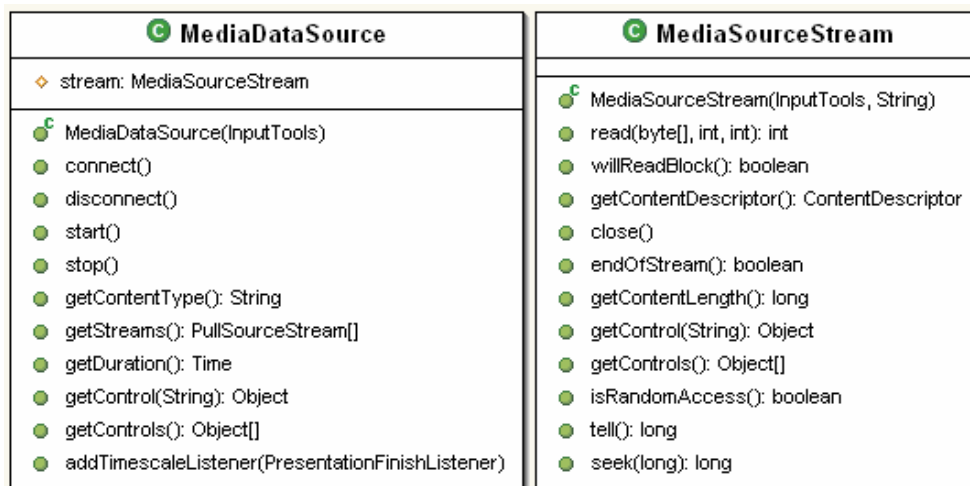


Figura 67 - Diagrama das classes da integração via programação com JMF.

As classes `MediaDataSource` e `MediaSourceStream` foram desenvolvidas para buscar dados de mídia em um buffer alimentado pelo algoritmo de ajuste elástico e fornecer a interface de uma fonte de dados definida pela especificação

<sup>12</sup> Disponível no próprio *site* do Winamp, ver .

JMF. A classe `MediaDataSource` herda das classes `DataSource`, `PullDataSource`, enquanto a classe `MediaSourceStream` herda de `SourceStream` e `PullSourceStream`. As classes herdadas são todas definidas pela biblioteca JMF. Originalmente, o exibidor de conteúdo JMF reproduz apenas fluxos de formato MP2 (Windows e Linux) e MPEG-1 vídeo (apenas no Windows). Entretanto, a adição da biblioteca `Jffmpeg` tornou possível fornecer suporte aos formatos MP3 e MPEG-1 vídeo (também no Linux). Vale destacar que o suporte de JMF a fluxos de bytes é diferente do suporte a arquivos.

Para realizar a comunicação via programação com o VLC é necessário utilizar JNI uma vez que esse exibidor está escrito em linguagem C++. O desenvolvimento dessa comunicação foi delegado para trabalho futuro.

Na comunicação com exibidores via RTP, os dados de mídia são enviados utilizando o protocolo RTP (Perkins, 2003). O transmissor escreve o fluxo de mídia numa porta e solicita que o exibidor de conteúdo escute essa porta. Os exibidores de conteúdo integrados foram o JMF e o VLC e a biblioteca para realizar RTP utilizada foi a JMF.

A biblioteca RTP do JMF permite apenas transmitir dados do MPEG-1 áudio formato MP2 e MPEG-1 vídeo e sistemas. Assim, embora o exibidor VLC suporte todos os formatos de arquivos manipulados por esse trabalho, só é possível exibir, utilizando o JMF para transmitir RTP, os formatos MP2, MP3 e MPEG-1. Entretanto, ainda é possível acrescentar novos *plug-ins* ao JMF para que este suporte novos formatos, tanto para transmissão via RTP quanto para reprodução. Por exemplo, quando integrado ao `Jffmpeg`, essa solução permite manipular também dados de MPEG-2 MP2 e MPEG-1/2 MP3.

A classe `TransmissionThirdPartPlayer` contém características comuns das classes `JMFRTPTransmissionThirdPartPlayer` e `VLCRTPPlayer`. Dentre outras atribuições, a classe `TransmissionThirdPartPlayer` é responsável por interagir com um servidor RTP para enviar os dados. A classe `VLCRTPPlayer` escuta a transmissão enviando um comando via um processo externa e a classe `JMFRTPTransmissionThirdPartPlayer` interage com uma classe cliente RTP para buscar os dados programaticamente. O transmissor e receptor RTP são implementados pelas classes `RTPServer` e `RTPCliente`, respectivamente.

### 5.3. Considerações Adicionais

A comunicação com exibidores de conteúdo pode ser também classificada como realizada de forma integrada ou não. Uma comunicação integrada possui duas vantagens principais: o fato do exibidor poder fornecer informações sobre o estado atual da apresentação da mídia (como, por exemplo, o instante atual da apresentação da mídia) e a possibilidade de posicionar espacialmente os exibidores de conteúdo dentro da interface gráfica da aplicação cliente.

As classes que realizam comunicação de modo integrado com exibidor de conteúdo implementam a interface `IIntegratedThirdPartPlayer`. O único exibidor de conteúdo cuja interação pode ser realizada de modo integrado é o JMF, via arquivo (tempo de compilação) e programação (tempo de execução). A classe `JMFIntegratedThirdPartPlayer` implementa os serviços definidos por essa interface e também agrupa um conjunto de características comuns entre `JMFFilePlayer` e `JMFDataSourcePlayer`.

Por fim, vale novamente ressaltar que atualmente estão disponíveis codificadores da SUN para poucos formatos de mídias. Para exibir formatos não suportados pelo JMF, as ferramentas VLC e Winamp podem ser utilizadas. Contudo, essas ferramentas executam de forma não-integrada com a aplicação cliente. O desenvolvimento de uma comunicação via programação com o VLC, já mencionado como trabalho futuro, pode permitir uma comunicação integrada com suporte a vários formatos.