

IPTV播放器客户端和服务端模块设计

袁明磊^{1,2}, 陈 业³

(1.安徽国防科技职业学院, 安徽 六安 221600;

2.六安大江信息技术有限公司, 安徽 六安 221600;

3.冠捷显示科技有限公司厦门创新中心TV部, 福建 厦门 361000)

摘 要: 在IPTV播放器的开发过程中, 一般用客户端、服务器端架构来实现。本文主要对IPTV播放器客户端和服务端模块进行了分析和设计, 为IPTV客户端和服务端开发提供了参考。

关键词: 客户端; 服务器端; IPTV

中图分类号: TP391 **文献标识码:** A

IPTV Player Client and Server Module Design

YUAN Minglei^{1,2}, CHEN Ye³

(1. Anhui Vocational College of Defense Technology, Lu'an 221600, China;

2. Luan Bigriver information technology Co., Ltd, Lu'an 221600, China;

3. TPV Display Technology Co., Ltd Xiamen Innovation Center TV Section, Xiamen 361000, China)

Abstract: In the development process of IPTV player, the client and server architecture often used. This paper focuses on the basic structure of IPTV player client and server modules, the IPTV client and server-side development to provide a reference.

Keywords: client; server; IPTV

1 引言(Introduction)

IPTV是一种利用数据进行音视频数据传输的新型媒体技术。IPTV相比模拟电视具有良好的互动性, 用户可以根据自己的兴趣进行内容选择。随着网络技术的发展, IPTV势必会成为未来家庭中的一种重要的业务形态。本文主要研究IPTV客户端和服务端模块的设计。

2 客户端模块设计(Client-side module design)

播放器客户端模块主要完成如下功能:

(1) 提供与播放相关的API供播放器应用层模块调用, 比如播放、暂停、跳转等。

(2) 封装来自播放器应用层模块的命令并以IPC的方式发送至播放器服务端模块。

(3) 接收来自播放器服务端模块的通知, 并通知给播放器应用层模块。

播放器在播放音视频时, 同一时刻播放音视频的个数主要受制于嵌入式设备上的解码器个数(软件解码和硬件解码)。一般情况下, 嵌入式设备上都会有且只有一个硬解码器存

在, 且为独占式的, 即有一个音视频在基于硬件解码正在播放时, 另外一个音视频是无法基于硬件解码器播放的^[1]。

但是, 在设计播放器架构时我们却可以做到预留添加软件解码器接口, 以便未来应对多路音视频同时播放的需求。而且, 播放器应该可以做到同时支持几个实例(相当于多个player)在运行, 比如在播放本地视频时, 一个播放器实例正在播放用户选中的视频, 此刻应该允许另外一个播放器实例负责获取本地文件夹下的其他视频的源文件信息(编码信息、播放时长、视频宽高、缩略图等), 这样在退出正在播放的视频时, 用户可以快速地预览到其他视频的源文件信息, 让用户感觉速度很快, 体验很好^[2]。

结合上述考虑, 客户端设计如图1所示。IMediaPlayer为MediaPlayerClient和服务端RealPlayer的共同接口; MediaPlayerClient为播放器暴露给应用层的与播放相关的接口; CmdWrapper用于打包来自应用层的与播放相关的命令; CmdSender用于向播放器服务端发送来自客户端的命令; EventQueue用于接受来自服务端的通知事件; PollThread用于

轮询EventQueue,并将每个Event派发给应用层。

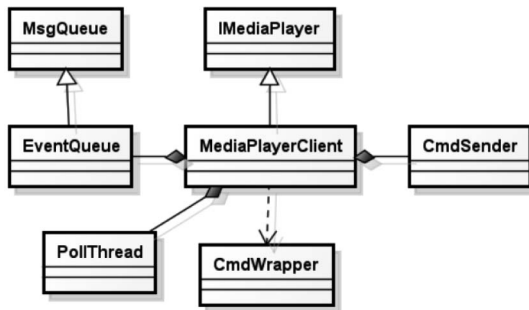


图1 播放器客户端设计类图

Fig.1 IPTV player client-side class diagram

3 服务端模块设计(Server-side module design)

播放器服务端模块主要完成如下功能:

- (1)接收来自播放器客户端模块的命令,并进行播放相关处理。
- (2)将播放器的相关播放状态通知给播放器客户端模块。
- (3)负责数据源获取模块、音视频分离模块、音频同步模块和音视频输出模块的调度和管理。
- (4)维护播放器的播放状态机。

播放器服务端的主要设计类图如图2所示。IMediaPlayer为MediaPlayerClient和服务端RealPlayer的共同接口;MediaPlayerServer为播放器服务端的代理接口,充当播放器服务端对外门户,来自客户端的命令都必须经过其转手派发给相应的某个RealPlayer实例,其内部保存一个RealPlaye实例链表;RealPlayer为播放器服务端核心管理类,与MediaPlayerClient一一对应,负责处理来自客户端的命令、维护播放器状态器切换、将播放器的状态以Event的方式通知给播放器服务端,同时,也负责MediaSource、MediaDemux、DecoderBase和Render之间的调度;CmdQueue负责接收来自客户端的命令;PollThread负责轮询CmdQueue并将命令派发给RealPlayer进行处理;EventWrapper负责打包服务端通知给客户端的Event;EventSender负责将Event发送给客户端;DecoderBase为解码器基类,HardwareDecoder为继承自DecoderBase的硬解码类,未来若是需要支持多路视频播放的话,只需从DecoderBase派生一个类,并实现相关接口函数即可;MediaDemux为所有媒体格式分离器类的基类;MediaSource为所有协议(网络或本地)的媒体源获取类的基类;Render负责对解码器解码完成后的音频帧和视频帧进行渲染输出。DecoderBase的某个派生类、MediaDemux的某个派生类、MediaSource的某个派生类和Render皆为MediaPlayerServer的成员变量^[3,4]。

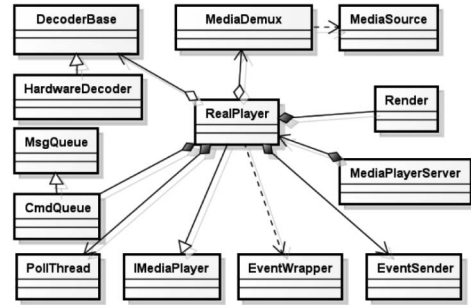


图2 播放器服务端模块设计主要类图

Fig.2 IPTV player server-side class diagram

故而,播放器客户端与服务端的交互设计如图3所示,播放器整体上采取了C/S(客户端/服务器)设计方式,即一个服务端进程可能会与多个客户端进程并存。应用层模块(LocalPlayer/Browser/...)和播放器客户端模块共处于客户端进程,与播放器服务端模块所处的服务端进程并存。当用户使用遥控器操作控制播放器的时候,操作命令会通过客户端模块发送至服务端模块;播放器服务端播放状态发生改变时,服务端首先会将通知事件发送至客户端模块,然后由客户端模块将其传送至应用层模块。

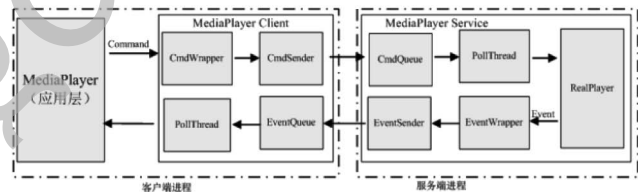


图3 IPTV 播放器客户端与服务端交互框架图

Fig.3 IPTV player client and server interaction frames

4 结论(Conclusion)

本文对IPTV播放器的客户端模块和服务端模块进行了设计。在实现客户端和服务端模块时,定义了播放器客户端和服务端的播放接口类。

参考文献(References)

- [1] 代黔豫.IPTV的核心技术和主要应用[J].移动通信,2009,33(2):84-87.
- [2] 李季.基于嵌入式Linux的IPTV机顶盒媒体播放器的研究[D].成都:天津大学,2008.
- [3] 徐晟华.基于MPEG-4的嵌入式流媒体播放器的设计与实现[D].镇江:江苏大学,2007.
- [4] 曾爱华.IPTV系统架构及主要技术[J].通信技术,2010,43(3):171-176.

作者简介:

表明磊(1985-),男,硕士,讲师.研究领域:计算机应用。