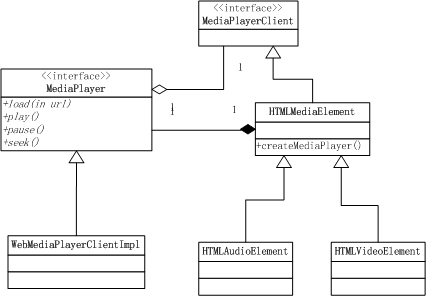
转载请注明出处: http://www.cnblogs.com/fangkm/p/3791964.html

在PC浏览器中播放视频,大部分视频网站都是采用flash播放器,这多亏了Adobe Flash Player庞大的装机量。但Flash Player对于浏览器来说就是一个插件,插件内部下载flv流、解析flv视频格式、播放视频等流程与浏览器都没有太大关系。浏览器新标准HTML5中增加了视频支持,如：<video src="movie.ogg"> </video>，video 元素当前支持三种视频格式：Ogg、MPEG 4、WebM。下面分析下Chromium源码中视频的播放流程。

视频播放流程有点复杂,下面还是从源头一点一点摸索。当blink遇到<video> tag时,会创建对应的Element对象HTMLVideoElement,同样遇到<audio> tag时会创建HTMLAudioElement对象.(其他tag对应的创建对象请参见core/scripts/make\_names.pl脚本生成的HTMLElementFactory.cpp文件)

HTMLVideoElement和HTMLAudioElement都是从HTMLMediaElement派生,接下来着重分析下

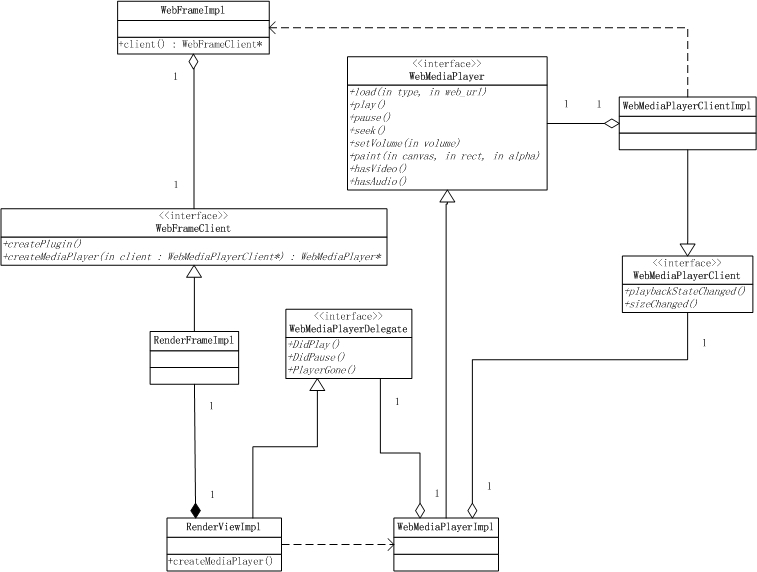
HTMLMediaElement类. 该类的结构图如下所示:



从结构可以看出,媒体播放逻辑抽象在MediaPlayer接口中, HTMLMediaElement的createMediaPlayer方法调用MediaPlayer的静态方法create来创建MediaPlayer对象，分析一下create的实现,其创建对象的逻辑依赖于由静态方法setMediaEngineCreateFunction

来指定，在初始化webkit的时候就通过该方法指定WebMediaPlayerClientImpl::create创建函数，从而最终创建WebMediaPlayerClientImpl对象来实现MediaPlayer接口逻辑。

然而WebMediaPlayerClientImpl也只是一层壳,它并没有真正的实现媒体相关逻辑,而是转接给WebMediaPlayer接口,相关结构如下:



WebMediaPlayerClientImpl通过依赖WebFrameImpl的WebFrameClient成员来创建WebMediaPlayer实现对象,通过WebMediaPlayerClient接口来响应WebMediaPlayer的相关事件, RenderViewImpl最终承接创建WebMediaPlayer的任务，其创建WebMediaPlayerImpl对象，并通过WebMediaPlayerDelegate接口来响应WebMediaPlayerImpl的相关事件。

下一篇文章开始分析WebMediaPlayerImpl部分, 这部分才真正开始音视频的相关逻辑。