

基于 LAsER 富媒体客户端的设计与实现^①

刘 麟^{1,2} 孙建伟² 林镜华³ (1. 中国科学院研究生院 北京 100049; 2. 中国科学院沈阳计算技术研究所 辽宁 沈阳 110171; 3. 中国科学技术大学 计算机科学与技术系 安徽 合肥 230027)

摘要: 随着移动终端用户量的增长, 用户对于体验的要求越来越高。在学术界, 富媒体技术以其丰富的用户体验成为了这一问题的解决方案。由我国国情决定, 富媒体技术尚处于初步研究阶段, 因此本文基于 MPEG4 LAsER 提出了完整的富媒体服务解决方案, 设计实现了一个以 SVG 为场景展现的富媒体客户端, 描述了系统分层体系结构, 分析了核心功能模块并对该系统进行了测试。实验证明, 富媒体客户端可以将各媒体资源有效的整合起来, 极大的增强了展现能力和实时交互能力, 丰富了用户的体验。

关键词: 富媒体; SVG; LAsER; 分层

Design and Implementation of Rich Media Client Based on LAsER

LIU Lin^{1,2}, SUN Jian-Wei², LIN Jing-Hua³

(1. Graduate University of Chinese Academy of Sciences, Beijing 100049, China; 2. Shenyang Institute of Computing Technology, Chinese Academy of Science, Shenyang 110171, China; 3. Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: With the increasing number of mobile end-user, people's demand of experience became higher and higher. In academy, rich media technology became the solution of this problem with its rich user experience. Because of the conditions of our country, rich media technology is still in the preliminary study stage, this article proposed a complete solution of rich media service based on mpeg4 laser, designed and implemented a rich media client with SVG scenes, described the stratified system architecture, analyzed the core functional modules and made a test to the system. The results showed that rich media client could integrat media resources very well, increased the demonstrated ability and real-time interactivity highly, enriched the user's experience.

Keywords: rich media; SVG; LAsER; stratified

1 引言

“富媒体”通常被定义为:“聚合多种媒体数据(例如音视频, 图片, 文字)动态、交互的体现”, 富媒体并不是像多媒体一样的一种具体的互联网媒体形式^[1], 而是指具有动画、声音、视频或交互性的信息传播方法, 富媒体的内容包括多媒体(二维和三维动画、影像及声音)。它包括 HTML、Java scripts、Interstitial 间隙窗口, Microsoft Netshow、

RealVideo, 和 RealAudio, Flash 等等, 涉及了比多媒体更宽的范畴, 包括编码、内容集成、流化处理、数字影院、用户端、数字版权保护和管理等多项内容。如在现实中, 它体现了多种媒体在保留其自身文件基础上的集成, 显示, 切换和网络流媒体广播, 用户端收看等。

MPEG-4 的第 20 部分是 MPEG 组织制定的 MPEG 标准^[2]中用于定义移动、嵌入式、消费电子的

^① 收稿时间:2010-04-07;收到修改稿时间:2010-05-06

富媒体标准，其中定义了两个标准：LASeR(Lightweight Application Scene Representation：轻量级应用场景呈现标准)和SAF(Simple Aggregation format：简单媒体聚合方式)^[3]。LASeR定义了场景描述语言和场景更新机制，SAF定义了富媒体内容的传输机制。其中，LASeR是一个开放的标准，可以利用有限的资源，如低带宽网络和低端终端，来展现丰富的媒体内容。该技术在SVG的基础上，提出了独到的场景更新、控制和资源传输方式。

本文对基于LASeR的富媒体客户端的设计与实现进行了阐述：首先介绍了富媒体客户端的方案设计，包括架构方案和系统设计，然后以基于SAF格式的场景流对该系统进行测试，并给出了分析测试结果。

2 方案设计

2.1 架构方案

为了实现基于富媒体的业务，除了客户端之外还需要富媒体服务器的支持，而且对于富媒体内容也需要制定相应的传输机制。一个完整的富媒体服务系统按功能逻辑可划分为富媒体服务器、传输机制和富媒体客户端3个大部分，如图1所示^[4]。富媒体服务器可分为富媒体内容(媒体场景、更新机制)、容器格式、传输包3大部分；传输机制包括两大部分，传输策略的选择(单播、多播、广播、流式分发)和远程交互机制的策略；富媒体客户端可分为富媒体播放器，本地交互和远程交互3大部分。富媒体媒体流首先由服务器发出，通过之前定义的传输机制(SAF)发送到富媒体客户端，用户通过与客户端的交互完成了整个过程。

该客户端采用分层体系结构，整个结构可以分为五层结构：其中最上层的是用户接口层，该层直接与用户界面打交道，直接和用户进行交互性操作，本地交互直接由客户端进行处理，远程交互需要与远程富媒体服务器进行互操作；接着由上到下依次是SVG展现层，图形元素树内存存储层，LASeR场景解析层和RTP、HTTP等协议接入层，如图2所示。这几个层次将在接下来的系统实现部分给出详细的介绍，根据其重要性的差别，对于SVG展现层，图形元素树内存存储层，LASeR场景解析层将进行重点的介绍。

客户端除了以上模块外，会需要用到会话控制机

制以及一些其他的配置信息的处理机制，这两个部分将作为两个单独的模块出现，本文主要讨论的是富媒体客户端的设计与实现，因此省略了会话管理与配置信息这两部分。

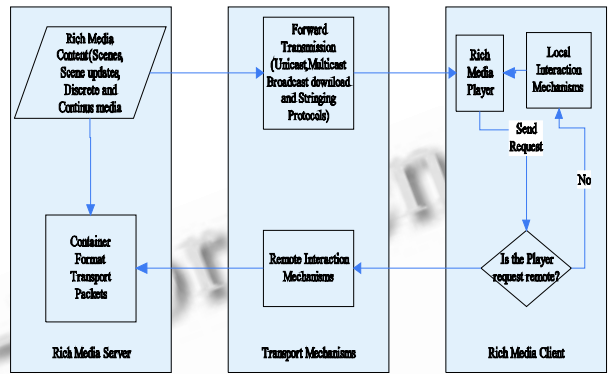


图1 富媒体系统结构图^[4]

2.2 系统设计

其中客户端富媒体播放器的系统设计可以分为之前介绍的5层，如图2所示，各部分的描述如下其中将重点介绍中间的3层：

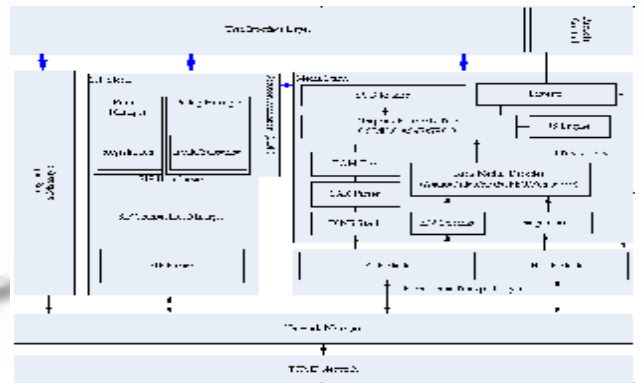


图2 富媒体客户端体系结构

1) 用户接口层，该层直接与用户界面打交道，直接和用户进行交互性操作，本地交互直接由客户端进行处理，远程交互需要与远程富媒体服务器进行互操作。其中本地交互需要处理以下几部分事件：DOM3 LEVEL事件，媒体访问事件，屏幕方向事件，其他事件。DOM3事件主要是基于XML文件格式的基本事件，包括基本的鼠标、键盘、定时、焦点捕获等事件。

2) SVG^[5]展现层，因为最终的富媒体的内容是以

场景的形式展现出来了，这里场景指的是包括各种富媒体元素，包括视频、音频、图片、文字等，然后将这些富媒体内容以 SVG 作为承载格式最终展现到客户端上，而 SVG 是什么呢？SVG 的全称是可伸缩矢量图形，这是一种全新的图片存储方案，不同与以往的图片存储方式，以往的图片是以像素的方式存放的，这样的话当图片在放大和缩小的时候就会出现失真的情况，而 SVG 是基于 XML^[6]文档的形式的，也就是说 SVG 是基于文本的，这样就具有很大的优势，比如将获得更高的图像质量、更精确的颜色、更灵活易用的文件格式、更加生动的表达方式、而且支持字符文字的查找(这在其他格式的图片格式中是不可思议的)，总的来说 SVG 是一种真正开放式的、不依赖供应商的 2D 矢量图形格式。

3) 图形元素树内存存储层，该层是 LAsEr 场景解析层和 SVG 展现层之间的接口层，起到承上启下的作用，该层首先接收 LAsEr 场景解析层传入的富媒体元素信息，包括 SVG 信息、视频、音频、图片和一些其他的 DOM 的规定要求等信息，将这些所有的内容通过适当的组织方式进行存储，这里采用的是树形结构的方式，其实从结构上来说该层可以不需要，可抽象出该层将使整个框架结构更加的清楚、分明，该层做的就是在内存中抽象出最终场景树，其实也可以说一个场景在内存中会有这样一棵树对应着。这样上层就只负责绘图的功能，不参与其他工作，这样层与层之间的耦合度就低了，方便于以后的扩展和修改。

4) LAsEr 场景解析层，该层主要解析从下层获得的富媒体信息，因为这些富媒体信息不是单独存在的，通过特定的传输机制，富媒体信息是以一定的包装形式传递的，这里就需要一个解包的问题，需要将下层收到的信息进行富媒体解包，解析过程与服务器端压包是逆过程，必须按照固定的格式解释，具体来说是以 XML 格式的文件进行的，因为整个富媒体场景都是用 XML 格式的文件表达的，因此在服务器端场景制作的时候，对于场景中需要的元素 SVG、音视频、图像文字都是以特殊的标记方式嵌入到 XML 文档中的，所以实际处理的时候就需要把这些信息提取出来，分类保存，这样才降低了耦合度，方便管理和处理。

5) RTP、HTTP 等协议访问层，该层所做的工作比较简单，某些图片和文字等信息就是简单的通过

HTTP 协议传送，不需要复杂处理。

3 系统详细实现

3.1 开发环境

本系统是在 Windows XP 操作系统上进行开发的，编程语言采用的是 java，选择 jdk1.5；IDE 集成开发环境选择的是 Eclipse3.5；客户端最上层表层采用的是 Batik，Batik 是以 java 语言为基础的，用来操作、处理和控制在 SVG 的一种软件工具集。

3.2 实现方式

接下来将详细介绍这个系统结构中最重要 3 层。

3.2.1 SVG 展现层的实现

SVG 展现层的采用 Batik^[7]工具集中提供的一个新的 Graphic2D 实现类，名为 SVGGraphics2D，在 SVGGraphics2D 中实现了 Graphics2D 一般 API 用于生成 SVG 内容，例如通过 drawRect，SVGGraphics2D 将生成对应的 SVG 内容并追加到 DOM 树中。最终，一系列渲染操作生成的 DOM 树表达了开发人员创建的图形，图形内容与其它实现完全一致，唯一的不同是用 SVG 描述的。如图 3 给出了 SVGGraphics2D 与 DOM 树之间的关系。

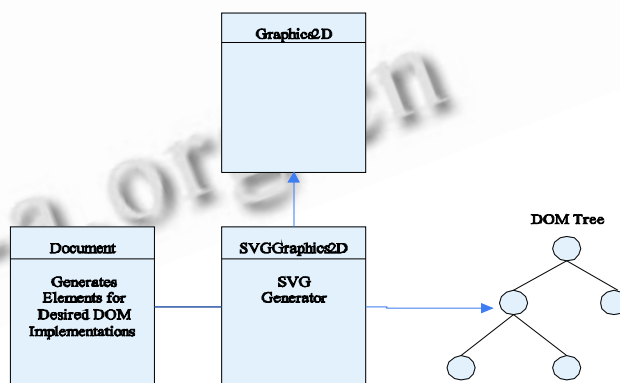


图 3 SVGGraphics2D 与 DOM 关系图

3.2.2 图形元素树内存存储层

首先图形元素树的内存存储层结构上是一颗树，想创建该树就必须定义树节点的存储结构，下面给出该节点数据结构并分析其功能：

```

Public class RootGraphicsNode extends
CompositeGraphicsNode{
    List treeGraphicsNodeChangeListeners = null;
    public RootGraphicsNode() {}
  
```

```

public RootGraphicsNode getRoot() {
    return this;
}
public List getTreeGraphicsNodeChange Listeners()
{
    if (treeGraphicsNodeChangeListeners == null) {
        treeGraphicsNodeChangeListeners = new
LinkedList(); }
    return treeGraphicsNodeChangeListeners;
}
public void addTreeGraphicsNodeChange Listener
(GraphicsNodeChangeListener l)
{
    getTreeGraphicsNodeChangeListeners().add(l);
}
public void removeTreeGraphicsNode
ChangeListener
(GraphicsNodeChangeListener l)
{
    getTreeGraphicsNodeChangeListeners().remo
ve(l);
}
}
    
```

其中 treeGraphicsNodeChangeListeners 是一个可以线性存储的 list 结构，从名字定义上就可以看出该结构是为每个图形节点保存了一组事件监听器。因为最后在整个内存图形树中每个节点都对应着最后场景上的某一元素，比如说一个节点对应这一个 SVG 的矩形图形元素，或者是文字、音视频等元素信息，而这些信息是需要和相关的事件对应起来的，比如鼠标移动到这些元素上，焦点的捕获和丢失，键盘等等各种事件。

接下来是对这时间监听器的添加、删除、获得操作。其实该图形节点已经是最高层的抽象了，可以看出它继承于 CompositeGraphicsNode，而该 CompositeGraphicsNode 又是继承于 AbstractGraphicsNode，AbstractGraphicsNode 中定义了几乎所有元素需要的细节。

3.2.3 LAsER 场景解析层的实现

LAsER 场景解析层的实现，依赖于其定义的媒体聚合方式 Simple Aggregation Format，简称称为 SAF。目的是可以将同一服务中的多个媒体数据高效复用成一股数据流，以避免终端从多方接收服务所需

网络 and 数据处理带来的系统消耗。SAF 与 LAsER 部分关系见图 4。

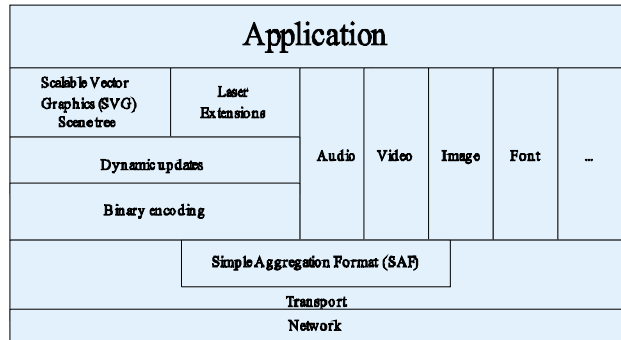


图 4 MPEG LAsER 技术架构图

为了使多条元素流在 SAF 流中同步高效的传输，SAF 定义了一个基本的实体叫 SAF Access Unit(SAF AU)，AU 又被封装在一个基本同步实体 SAF packet 里，由此解决了场景内容的实时性，时效性，各种媒体的同步，聚合等问题。

SAF SL Packet Header	SAF SL Packet Payload	
SAF Packet Header	SAF AU Header	SAF AU Payload

图 5 SAF packet 数据包结构图

SAF access unit 是 SAF 带有时间信息的最小传输单元，也是最小可访问单元。一个 SAF Access Unit 包含一个两字节的头部(SAF header)和一个字节序的负载(SAF Payload)。SAF 流是由多个 SAF access unit 组成的。

在具体实现场景解析层的过程中，解析过程是场景压缩过程的逆过程，按照 SAF header 中定义的不同信息，将各种不同的媒体信息进行分类的收集整理。

4 系统运行测试结果分析

本富媒体客户端并不是一个特别复杂的系统，但其场景描述语言支持的元素众多，每个元素又有相当多的属性；并且整个引擎中还有很多模块需要协同工作，尤其是 DOM，仅这个模块就需要 45 个接口，每个接口又包含几个到十几个不等的方法。而且如果进行测试还需要富媒体服务器的配合，考虑到这些因素，这里我们采用本地模拟远程服务器的方式。

测试采用对一般节点、文字节点和图形节点分别完成测试。测试一般节点主要是对 SVG 的基本图形元素进行测试,其中矩形分为直角矩形和圆角矩形两种,所有节点的 fill,stroke 属性均设置相同,主要测试一下属性 line、polygon、rect1_no_rxy、path、rect2_rxy、circle、ellipse、image。测试文字节点主要对文字节点 text 和 textArea 进行测试,其中所有节点的属性均设置为相同。测试图形节点主要针对 image 进行测试,其中所有节点的属性均设置为相同(x=" 0" y=" 0" width=" 256" height=" 103" xlink:href=" test.png")。最后测试结果如图 6。

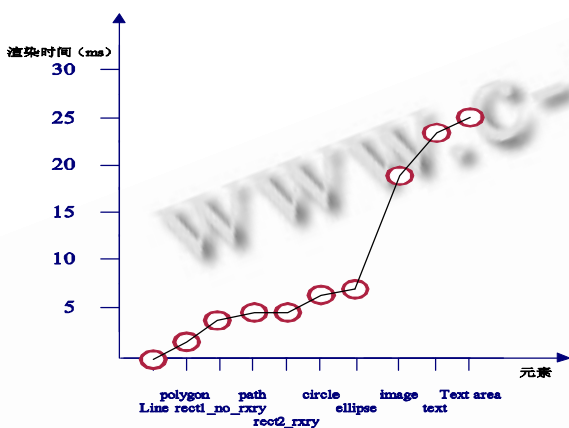


图 6 各场景元素渲染时间特性图

表 1 性能参数

参数	指标	
CPU 频率	加速模式 600MHZ	23Fps
	正常模式 512MHZ	12Fps
	省电模式 360MHZ	7Fps
ROM	宋体 gb2312	580kbyte
	Jpeg svg png lib	380kbyte

在移动终端环境上(HP nx6330),根据对测试数据的测试统计,引擎的各性能参数见表 1。由测试数据可以看出,引擎已经能够支持上面定义的各项描述标签,渲染效果比较理想,就是最复杂的图形元素,渲

染时间最多也就 25ms。在 545M 主频的高端嵌入式设备中,平均运行可以达到 23Fps 左右,就算是运行在 198M 主频的低端设备上,最低也可以保证 13Fps。

5 总结与进一步工作

本文从提出一个富媒体客户端的框架方案到设计各模块的功能,最后实现该客户端的各模块功能并测试了该客户端。实验证明,富媒体客户端可以将各音视频及动画图像等资源媒体有效的整个起来,极大的增强了展现能力和实时交互能力,丰富了用户的体验。

目前,富媒体技术的使用主要集中在 PC 环境中,和音视频的整合,与其他媒体平台如游戏、无线、电子杂志等领域的合作,将成为富媒体技术的发展趋势[8]。伴随着 Web 与 3G 技术的发展,富媒体数据源将会融合广电数据广播网络和移动网络。如何适配这些传输网络,以及在富媒体场景包含大量信息情况下,如何保证用户流畅观看内容的同时保证场景实时,动态的推进,是进一步工作的方向。

参考文献

- 1 王琦. Peter J.Brown.富媒体提供的机会.卫星电视与宽带多媒体, 2006,21(9):297 - 300.
- 2 Dufourd JC, Avaro O, Concolato C. An MPEG standard for rich media services Multimedia, IEEE 2005,12(4):60 - 68.
- 3 ISO/IEC 14496-20. Information technology - Coding of audio-visual objects-Part 20: Lightweight Application Scene Representation(LASeR) and Simple Aggregation Format(SAF) 2006.
- 4 3GPP TS 26.142 V7.1.0(2007-09)Technical Specification 3rd Generation Partnership Project;Technical Specification Group Services and System Aspects; Dynamic and Interactive Multimedia Scenes; (Release 7).
- 5 W3C, Scalable Vector Graphics (SVG) Tiny 1.2.
- 6 W3C Document Object Model(DOM)Level 3 Events Specification,Version 1.0,W3C.
- 7 Kolesnikov A. Java Drawing with Apache Batik:A Tutorial, 2007.
- 8 蔡安妮,孙景鳌.多媒体通信技术基础.北京:电子工业出版社, 2000.