

数字法庭关键技术的研究与应用

摘要

随着我国各项法律制度的不断完善和健全,对庭审过程公开、公正高效、真实透明的要求及民众呼声日益增多,为了适应社会形势和经济法制建设发展的需要,加快科技强法步伐,实现公正裁判,提高法院社会公信度及法官自身素质和审判水平,充分体现阳光审判的原则,利用科技手段推动司法文明建设的进程,完整、准确、实时数字加密地记录、再现庭审理听过程,避免当事人各方事后对笔录内容和庭审过程的争议,是保证司法公正的重要科技手段之一,符合我国现行法律对庭审记录要求的基本精神。数字法庭正是基于上面的要求下产生的。

本文主要对数字法庭的两个关键技术进行了描述:复合笔录技术和视频点播直播技术。复合笔录技术一种实现文字与视频联动的方法。在现在的法庭审理过程中,书记员会做笔录,在法庭里安装的视频音频设备会记录现场的声音和画面。通过再现庭审理听过程,避免当事人各方事后对笔录内容和庭审过程的争议。通过本技术可以实现在 word 笔录文本里双击任意处的位置,播放器随即自动定位到与文字相对应的视频位置,并开始播放,同时还实现了拖动视频到任意位置,word 即定位到相应处文字,从而达到文字视频互动的目的。视频点播技术是指用户可以在任何时刻从服务器(群)中获取所需的任意长度的音、视频数据到客户端播放的应用,是一种新颖的交互式多媒体服务。

上面的两个关键技术应用到了实际的数字法庭系统开发中,通过测试,取得了很好的效果,现在已投入试用。

关键词: 数字法庭系统 复合笔录 视频点播 视频直播

THE RESEARCH AND APPLICATION OF DIGITAL COURT KEY TECHNOLOGIES

ABSTRACT

Along with the legal systems improving, the sound of the trial process that will be open, fair and efficient, transparent and the real requirements are increasing. In order to adapt to the situation of social and economic development of the legal system, science and technology to accelerate the pace of strong laws to achieve Magistrate Court justice to raise the social credibility and judge their own quality and level of the trial, the trial fully reflect the sun principle, the use of scientific and technological means to promote the judicial process of civilization, complete, accurate, real-time encryption and digital records, audio-visual reproduction process of the trial, the parties to avoid The parties after the trial transcripts and the contents of the controversial process, to ensure that justice is an important means of science and technology, more in line with the existing Chinese laws on the trial record required by the basic spirit.

This article on the two key technologies of the court were described complex record and a live video-on-demand technology(VOD).Composite record of technology referred to as video-on-demand VOD, is the Video On Demand in the English abbreviation, it means that users can at any time from the server (group), get access to any length of the audio and video data to the client applications Player is a novel Interactive multimedia services. Video (Video) contains a lot of images, text, and other media format simply can not provide the content, to be able to provide users with more and more information.

Above the two key technologies have been applied to the actual court systems, through testing, achieving good results, and now having been put into trial

KEY WORDS: digital court system ,composite record, video on demand, video broadcast

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 李鹏宇 日期： 2009.3.19

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在__年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名： 李鹏宇 日期： 2009.3.19

导师签名： 邱中亮 日期： 2009.5.19

第一章 绪论

1.1 研究背景

数字法庭是指围绕法庭开庭审判,利用计算机网络、AV集成、多媒体、数据库等先进技术和产品,为书记员、法官和双方当事人等参与诉讼提供辅助支持,为法庭举证示证提供多种展示平台,为完整、翔实记录庭审全过程提供多种记录手段,为法官开庭审判提供方便、快捷的集中控制功能,从而达到听得清楚、看得明白、记得完整、用得方便的效果。

国家“十一五”规划期间《人民法院物质建设规划》关于信息系统在审判管理中的应用要求明确提出,各级法院要在2009年底前实现案件的实体管理,实现上诉案件和复核案件的电子卷宗报送、异地质证、远程庭审观摩等。伴随着我国司法改革的进一步推进,依托数字法庭的先进技术,更多的维护诉讼参与人权益的措施将会不断地出现。数字法庭对法院内部来讲,实现了审判全过程的动态管理与监控;对社会和当事人来讲,就是在法律许可的范围内,满足当事人的知情权,让公正在彻底的透明中实现。

信息化、科技化的管理已经成为社会发展的趋势,信息化是一个地区和单位综合实力和文明程度的重要体现。现代信息技术的发展与应用同样为法院的现代化建设提供了良好机遇。在这样的背景下,加强法院的信息化建设,可以说是应时而动。现代信息化手段在人民法院的广泛应用,不仅可以使法院管理水平实现新的飞跃,还可以增进资源共享,提高法院信息传递速度,为法院“公正与效率、司法为民”的工作主题提供重要的技术支持,更可以节约诉讼资源,方便群众诉讼,进而有效提升司法权威和司法形象。

1.2 数字法庭系统的发展和课题的意义

1.2.1 数字法庭系统的发展

自1996年修正后的刑事诉讼法对庭审方式进行改革,即确立控审分离、控辩对抗为基点的庭审规则之后,各级法院逐渐开始试行用科技手段支持庭审工作。这些年,全省法院的审判法庭数字化工作一直处于自由发展的状态。从设施构成看,主要出现了二种形态的科技化法庭。

第一种，初期形态，主要由庭审记录设备、实物投影仪和音响系统组成，各法院习惯上称之为“多媒体法庭”。

第二种，改进期形态，是在“多媒体法庭”的基础上，引入了视频会议系统，以支持对庭审实况的录制、转发，但各种设备基本上是相互独立的，仅有部分实现相互联动，如由扩音系统控制摄像机，各法院习惯称之为“智能化法庭”。纵观这两种形态的科技化法庭，基本上处于设备的操作使用层次，没有形成一个有机的整体。虽然满足了部份实际应用需求，但缺陷很多，具体体现在：

无法支持电子证据的有效举证。

庭审与会议有着不同的特点，视频会议系统不能很好地解决庭审应用需要，如，网上图像应用问题。

缺少信息网络系统的支持，各种设备相互独立工作，所产生的信息不能互通互用。

功能固定化，没法适应庭审需求的变化。

为规范各级法院的审判法庭数字化工作，整合法院科技化应用平台，疏通法院业务的信息流程，完善审判法庭科技支持手段，最大限度地满足审判实际需要，提高信息资源共享度和庭审效率。

1.2.2 数字法庭系统的意义

1. 解放书记员

在数字法庭系统建设前期，法院主要是在法庭内安装一些电子设备，如会议音响、摄像机、大屏幕显示、实物展台、电脑、光盘刻录机等，系统主要特点是简单满足庭审过程中审判人员和诉讼参与人的视听感受和部分新型资料的证据展示的需要，同时实现了庭审过程的光盘刻录后随案卷保存。随着电子化设备的增加，书记员操作和管理设备的工作量和复杂度也在增加，同时庭审实体信息以原始介质分散保存，不方便后续利用，无法发挥庭审信息采集后的利用价值。

为了解决电子设备的操作和管理难题，把书记员从繁重的操作中解放出来，系统建设后期应在简单电子化的基础上增加设备的集中控制系统，并根据庭审过程的需要进行设备联动。集控系统的应用大大提高了法庭支持环境的可管理性和可操作性，提高了各类电子设备的可用性，降低了书记员的工作复杂度，从而使得书记员可以更好地集中精力进行庭审笔录。

2. 有效规范法官庭审行为，通过众多高科技手段的有效应用，提高法官工作效率的同时也有助于提升庭审质量。

3. 为可能发生的争议提供原始资料，保护庭审各方及一线庭审工作人员的权益。

4. 实现网络听审、远程举证，使庭审工作不再受到时间、地域的限制。

5. 为考核法官提供客观详实的声像资料。

1.3 数字法庭系统研究重点

数字法庭组成主要包括法庭庭审管理软件、法庭支持环境和集控系统、采集与存储系统、音视频直播点播系统和系统平台等。站在审判业务应用支持和辅助的角度，数字法庭的功能定位主要包括如下几个方面：

1. 提供举证证证功能，满足不断增长的新需求。通过配置多媒体证据展示设备和计算机终端，展现法律规定的各种类型的证据，支持法庭辩论和案情讲解与演示，使双方当事人、旁听人员、合议庭成员对案件案情有直观、切身和全面的感受，也可以通过计算机网络技术实现异地质证，解决举证难、证人作证难等问题。

2. 实现文字、音频和视频等多种格式的数字记录保存，并且可以通过回放再现庭审过程，提高审判透明度。对大案、要案，社会影响大、案情复杂、法律适用不太明确或者新型案件的庭审，通过运用视频及其处理技术，可以将庭审过程中各个阶段的图像和声音完整记录和存储下来。这一方面为庭审笔录的准确性提供校对基准，另一方面还可以通过视频点播等方式再现庭审过程。庭审过程再现可以为法院提供审判监督、案例参考、法官考核、案例学习等应用，进一步提高了审判的透明度，便于社会公众对审判过程进行监督。

3. 实现当庭结案案件裁判文书的当庭送达，减少当事人的诉累，提高工作效率。目前人民法院审理的案件中，民事案件占总数的 70% 以上，对事实清楚、证据确凿、罪刑轻微的刑事案件和事实清楚、权利和义务关系明显的民事案件，系统支持当庭结案、文书制作、签章打印、当庭送达等功能，达到便民诉讼的目的。

4. 实现庭审流程管理与控制的信息化管理，提高庭审管理能力。从庭前准备开始，案件审理一般要经过开庭、法庭调查、辩论、质证、休庭、闭庭等几个阶段。系统为审判长、书记员等法院审判人员提供信息化工具，满足各个角色在案件审理不同阶段的信息获取需要和信息展现需要，如案件基本信息、庭审笔录信息等。

5. 庭审笔录及校对。书记员可以进行庭审笔录与校对，校对时可以对庭审录像进行回放，校对完成后可以当庭打印笔录。

6. 视频直播点播功能一方面为远程审判、异地质证、脆弱证人保护等审判业务应用提供支持，另一方面可以进一步扩大庭审公开的范围，提高庭审的透明性和公正性。

7. 集中化的系统管理与操作，减轻系统维护。数字法庭中涉及的设备很多，各种设备的操作也可能比较繁琐。为提高审判效率，需要根据案件庭审及其庭审流程的需要，将法庭内各个设备的操作与管理简化，尽可能减少诉讼参与人的设备操作。数字法庭应根据庭审流程的需要，实现设备之间的智能联动。

8. 数字法庭系统与审判信息管理系统结合以后，一方面，数字法庭系统可充分

利用审判信息管理系统中的案件信息;另一方面,数字法庭产生的庭审音视频信息、庭审笔录等信息可与音视频直播点播系统、审判信息管理系统、电子档案系统的信息等实现共享,为系统应用奠定基础。

1.4 论文的主要内容

论文共分为五章。组织如下:

第一章为绪论,主要介绍了课题研究的背景和意义,主要研究内容以及本文的组织结构。

第二章为数字法庭系统概述,主要介绍了数字法庭系统的框架,功能。

第三章为数字法庭关键技术的研究,主要介绍了数字法庭系统的几个关键技术。

第四章为数字法庭系统关键技术的应用,主要介绍了关键技术的实现及相关测试。

第五章总结数字法庭系统的特色,分析现有的问题,提出进一步优化的设想和展望。

第二章 数字法庭系统概述

2.1 数字法庭系统构架

庭审活动是以法官为中心，原、被告及其各自的律师、证人参与的交互活动。从审判业务的角度来讲，数字化审判法庭是指借助于现代科学技术装备，通过扩展、延伸传统的审判法庭功能，适应新的庭审需求，达到强化庭审效果、提高庭审效率、促进审判工作的公正、公开的现代化审判场所。

从 IT 专业角度讲，是一个集语音、数字、图像处理为一体，综合应用数据库技术、网络技术、自动控制技术的真正的多媒体系统。通过构建省高院、各中级人民法院、专业法院和基层人民法院的庭审数字影像网络系统，我们要达到以下目标：

1、打造一个功能完善、操作便捷、适用于各种证据类型的庭审证据展示系统。通过这一系统，规范举证程序，提高举证的效率，增强电子证据举证的可视性，使旁听席上能听得清楚、看得明白。当庭采录的证据可进入计算机网络系统的证据库，实现各类证据的数字化，为案件卷宗的数据化奠定基础。

2、构建庭审支持信息网络系统。与现有的审判信息管理系统实现无缝联接，使庭前准备信息、当庭产生的信息、庭后合议信息实现网络化处理，最终形成多媒体电子卷宗。

3、构建可视化典型案件案例库，实现典型案例庭审实况的数字化采录、存储、查询、播放，为审判研究、交流以及教学培训提供直观、详实的原始资料。

4、构建全省法院系统庭审观摩网络，实现各级审判组织对疑难复杂案件庭审的实时观摩，增强审判研究的客观性，提高审判指导的针对性。

5、构建网上直播平台，有选择地向社会公众直播一些有重大社会影响案件的开庭实况。使百姓有更多机会观摩法院开庭，利于增长市民法律常识，增强自律意识，利于法治建设。

6、在广泛实践的基础上，逐步试行证人远程网上作证，解决特殊情况下的证人出庭作证问题（出差在外，不在法庭所在地、不愿公开露面作证）。待外部条件成熟时，与检察院、拘留所、监狱进行视频联网，试行刑事案件网上开庭，提高开庭的安全性，降低司法成本。

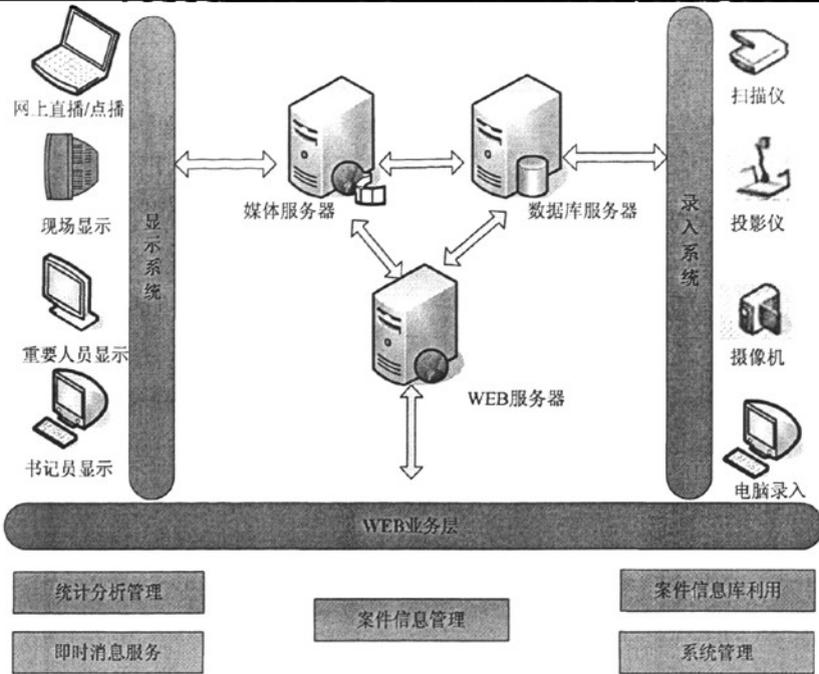


图 2-1 数字法庭系统构架图

2.2 数字法庭系统功能介绍

数字法庭庭审录播系统应具有的基本功能：

- 1、语音激励视频跟踪功能，保证整个庭审过程录像的完整性和流畅性；
- 2、庭审影像视频参数调整功能，保证在不同的环境下，庭审录像质量最优；
- 3、多画面同屏浏览人工编辑视窗功能，保证庭审过程画面的完整性；
- 4、庭审理听资料重点段落即时标识功能，保证对庭审过程中当事人、法官对任何重点的发言，举证进行；
- 5、庭审理听资料片头文字叠加功能，保证庭审信息的完整性；
- 6、视听资料当事人视频真迹签名确认功能，保证庭审录像的真实性和有效性；
- 7、DVD 光盘数字加密实时直录功能，保证视频资料的法律有效性，能够作为司法证据进行保存和发布；
- 8、庭审理听资料库存储、备份功能，保证庭审理听资料的数据安全性；
- 9、庭审网络直播、点播功能，保证法院系统内，在有授权权限的情况下，进行案例分析和案例讨论；
- 10、能够将数据保存入全省统一的电子档案系统内，保证庭审录播的影像文件成为河南省省法院资源数据库重要的组成部分。

庭审活动中法官通过其法台上的计算机，可以：

- 1、看庭审活动预案、相关案情资料；
- 2、看相关法律法规文件；
- 3、查看相关案情的影像资料；
- 4、查看当庭的实况影像；
- 5、查看异地法庭的实况影像，实现异地开庭和远程作证；
- 6、查看当事人各方计算机举证内容；
- 7、查看实物展台的举证内容；
- 8、控制上述影像和举证内容在庭内显示设备上同步输出；
- 9、能够和后台进行实时的信息沟通；
- 10、能够查看和确认参与庭审的当事人身份信息；
- 11、控制庭内灯光、窗帘、门禁、原被告双方的话筒及投影等设备；
- 12、将上述影像和举证内容采录入电子档案软件，作为庭审卷宗数据的一部分进行保存；

庭审活动中书记员通过其书记员席上的计算机，可以：

- 1、完成庭审前准备；
- 2、对参与庭审参与人员的身份信息确认，并保留身份证扫描信息；
- 3、速录庭审活动；
- 4、通过庭内显示设备与当事人核对庭审速录；
- 5、在法官授权的情况下协助法官完成法官相关的上述控制功能；

在原、被告席上设有网络和设备接口，庭审活动中原、被告在法官的允许（控制）下，通过其自带的计算机，可以：

- 1、在英特网上获取相关资料；
- 2、将准备好的资料（文字、图片、录像）通过庭内显示设备输出；

庭外的有关人员在获得授权的情况下，通过法院内部的计算机网络或法院系统专网，可以：

- 1、查看相关案情的文字、影像资料；
- 2、查看庭审的当庭实况直播；
- 3、查看庭审的当庭举证内容；
- 4、查看相关庭审实况的录像；
- 5、与法官进行信息交流；

因此，数字化审判法庭的功能描述如下：

1. 影像采集与网络直播

将各路摄像机的视频信号、混音器的音频信号输入到庭审管理中心中的服务器，服务器首先对输入的音视频信号进行数字化，之后采用先进的音视频数据压缩

算法将其转换成具有满足 MPEG-4 标准的码流, 以组播或单播的形式发送到计算机网络上, 经授权的用户在连入法院专用网络的计算机上使用浏览器收看庭审现场实况。

当事人的举证信息、示证内容经过系统的处理也可以在网络上同步直播。

在网络管理员的支持下, 可以有条件的在互联网上直播。

2. 数字影像存储

庭审实况的数字影像(音视频数据)采用分布式存储与集中存储相结合的方式, 在庭审管理中心上或专门配置的大容量存储服务器上存储并自动归档。当网络相对空闲时, 系统可自动将各个庭审管理中心上存储的数据通过网络自动传送到中央大容量存储服务器(磁盘阵列)上。

系统将同一庭审活动的几路影像和举证内容自动合成一个完整的庭审活动录像资料, 并可进行智能分类检索。

大型存储设备在选型是应采用有 IP 接口的存储设备, 要求数字法庭存储设备全省 IP 可达, 在存储结构上应该采用成熟 SAN 架构, 并为了便于远程数据共享, 支持点播, 要求支持 NAS 功能。

3. 数字影像点播

庭审活动录像资料点播是指经授权的用户可能通过智能检索系统调阅某一庭审的录像资料。用户无须了解各个庭审活动录像资料的存储位置, 用户所面对只是智能检索系统, 用户输入其所要检索的庭审活动的关键字, 系统会列出符合检索条件的庭审活动录像资料列表, 用户先选择相应的庭审活动的录像资料, 再选择播放形式。之后系统会根据用户的选择, 向点播服务器请求点播所需的庭审活动的音视频资料, 点播服务器根据系统内的存储记录, 取得相应的音视频资料, 以组播或单播方式将用户点播的庭审活动的音视频数据发送到网络; 之后用户使用专用的流媒体播放软件收看其所点播的庭审活动录像资料。

点播庭审资料时, 系统要保证同庭的多路影像、举证信息、远程举证在播放时同步。

4. 庭审数据资料管理

庭审数据资料管理包括文字数据资料管理及相关的庭审活动的影像资料管理, 它提供对庭审数据影像资料进行存储和备份服务。具体表现为法院庭审活动自动录像、智能编辑、自动归档、智能检索回放能力。

庭审数据资料管理必须能够与河南省法院电子档案管理系统无缝连接。即庭审数据资料管理能够自动转入河南省法院档案管理信息系统内, 电子档案系统里可以直接调阅庭审数据影像资料。

5. 电子举证

包括实物举证投影、当事人自带电脑举证、网上证据引用、远程证人举证等功

能。

系统在实物举证投影的同时,可以将实物举证的图像数字化并成为庭审影像资料的一部分。

系统允许当事人自带电脑运用任何技术手段用电脑屏幕显示的内容(可视)支持自己的举证。系统可以将当事人自带电脑的屏幕显示内容在庭内显示设备上显示、同时在网络直播并采集成为庭审影像资料的一部分。同时,系统必须保证当事人自带电脑不能与法院内部网络连接,也不能允许当事人自带程序在法院电脑中执行,以保证法院网络免遭攻击。

网上证据的引用也必须采用不与法院内网连接的电脑(或当事人自带电脑)通过系统进行,以确保法院网络的安全。

6. 电子显示

法庭内设有电子显示设备以保证各种证据的清晰显示。在每个法官、原告席和被告席、证人席、犯罪嫌疑人席设内嵌式液压升降液晶显示器,法官、原告、被告引用的各种文字、图片、影像资料均可以在法官的允许(控制)下在显示器上输出。为了法庭的旁观席能够清晰观看原、被告和证人、犯罪嫌疑人的图像及其举证材料,在旁观席前方应设有大屏幕投影或大屏幕等离子显示器。法庭外设有大屏幕显示,庭审安排、公告信息、公开的庭审实况均可以经过授权在其上显示。

7. 异地举证和网络开庭

异地(庭外)证人可以使用系统进行举证,异地(庭外)证人通过网络将声音和图像传入庭内并可以和法官及原、被告双向对话,实现远程举证。法官可以控制异地证人的影像是否投影输出,并可在必要的时候对证人的影像做局部遮挡处理、对证人的声音做变音处理,以实现不宜公开出现的证人的隐蔽作证。

分布于不同地域的几个法庭可以在同一时间通过网络进行庭审活动。各个法庭内的各个显示屏在同一时间显示所有参与庭审活动的各个法庭的现场活动影像,从而使得分布于不同地域的各个庭审元素可以实时协同完成一个完整的庭审活动。

8. 集中控制

法官或者法官指定的法院工作人员可以通过授权的网络终端对各个庭审元素

2.3 数字法庭布局

数字审判法庭布局应如下图所示:

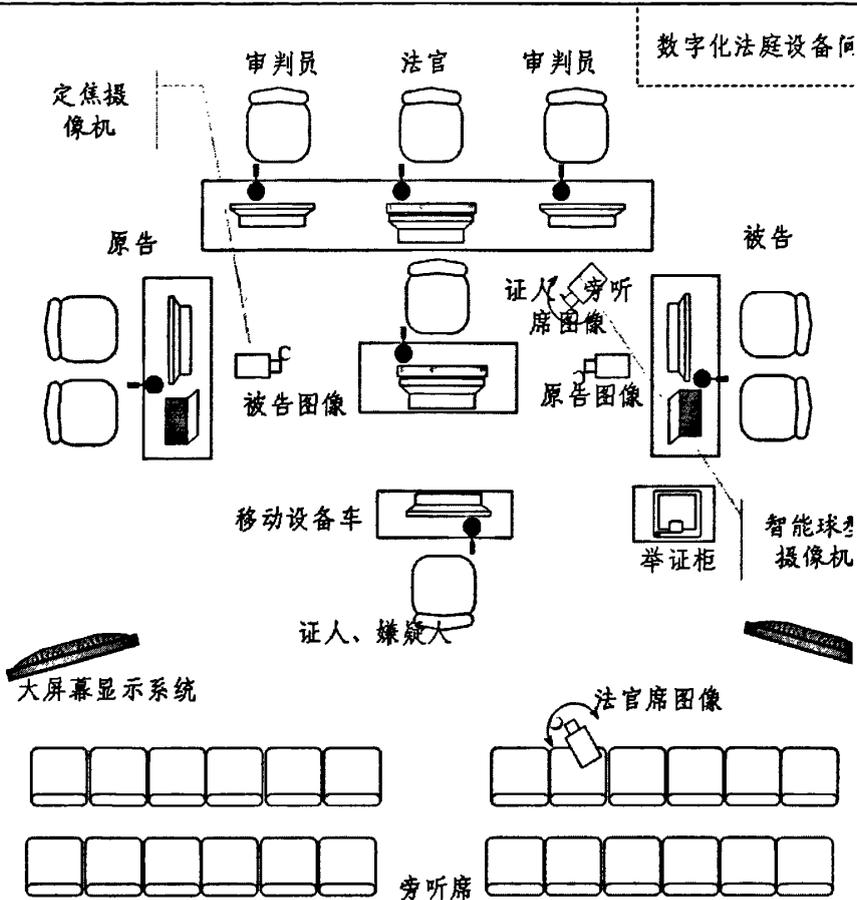


图 2-2 数字审判法庭总体布局图

数字化审判法庭的信息布局可分为计算机网络、音视频采集、举证信息采集及庭上设备控制四大部分。

1、计算机网络：计算机网络采用三级体系结构。

(1) 法庭上法官、书记员及法院工作人员的计算机、显示扩音控制系统、流媒体服务器和庭审智能控制服务器都接入到法庭局域网络。

(2) 法庭局域网络通过网络连接设备接入法院专用网络，网络连接设备可以控制该法庭的音视频数据流是否发送到法院专用网络，并可控制远程终端是否可以接入该法庭局部网络。

(3) 法院专用网络内的内容，通过隔离设备（网闸）转入因特网。

三级网络可以充分保证法庭上的各个用户、法院专用网络及远程网络上的已认证用户之间进行实时安全的信息交互。

2、音视频采编播：

法庭上各个摄像机和拾音器采集的视音频信号通过庭审管理中心压缩加密之后以组播方式发送到法庭局部网络上，法官可以通过庭审管理中心上的控制器来选择某路图像在庭内显示设备上显示。

3、举证信息采集编播：

法庭上各个举证设备如从实物投影仪、当事人自带电脑的显示、录像机等采集的视音频信号通过庭审管理中心压缩加密之后以组播方式发送到法庭局部网络上；法官、书记员、原被告及其辩护律师的计算机和实物展台的图像显示信号接入庭审管理中心，法官可以通过控制庭审管理中心来控制选择某路具体物证信息在庭内显示设备上的显示。

4、庭上设备控制

法庭上的各种设备如投影仪升降系统、幕布控制器、窗帘控制器、灯光控制系统、法庭门禁系统、报警器及传唤系统等设备通过总线连接到智能设备控制服务器上，智能设备控制服务器通过法庭局部网与法官的计算机相连；法官或法官授权的法院工作人员就可以在自己的计算机上控制庭上各种设备。

第三章 数字法庭系统关键技术的研究

3.1 Java Struts 技术概述

著名的软件大师 Ralph Johnson 对框架(Framework)进行了如下的定义： 框架是整个系统或系统的一部分的可重用设计，由一组抽象的类及其实例间的相互作用方式组成^[1]。

框架一般具有即插即用的可重用性、成熟的稳定性以及良好的团队协作性。J2EE 复杂的多层结构决定了大型的 J2EE 项目需要运用框架和设计模式来控制软件质量。目前，市场上出现了一些商业的、开源的基于 J2EE 的应用框架，其中主流的框架技术有：基于 MVC 模式的 Struts 框架和基于 IoC 模式的 Spring 框架以及对象/关系映射框架 Hibernate 等。

1. 表示层框架 Struts

Struts 是一个在 JSP Model2 基础上实现的 MVC 框架，主要分为模型(Model)、视图(Viewer)和控制器(Controller)三部分，其主要的设计理念是通过控制器将表现逻辑和业务逻辑解耦，以提高系统的可维护性、可扩展性和可重用性^[2]。Struts 框架的体系结构如图 3-1 所示。

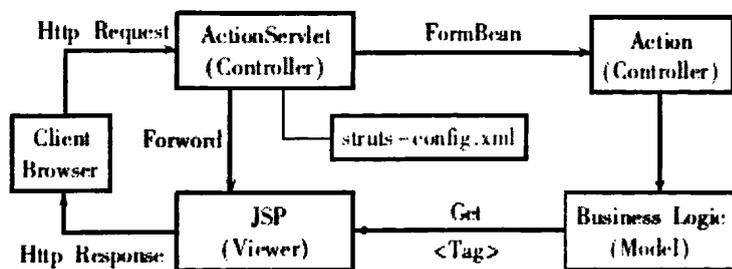


图 3-1 Struts 体系结构图

下面就图 1 所示的体系结构图分析 Struts 框架中的 MVC 组件。

(1) 视图：视图部分主要由 JSP 页面组成，其中没有流程逻辑、业务逻辑和模型信息，只有标记。Struts 自身包含了一组标记库(TagLib)，这也是 Struts 的精华之一，灵活运用它们可以简化 JSP 页面的代码，提高开发效率。

(2) 控制器：Struts 中的 Controller 主要是其自身提供的 ActionServlet。

ActionServlet 接收所有来自客户端的请求并根据配置文件(struts-config.xml)中的定义将控制转移到适当的 Action 对象。

(3) 模型: Struts 没有定义具体 Model 层的实现, Model 层通常是和业务逻辑紧密相关的, 有持续化的要求。目前在商业领域和开源世界, 都有一些优秀的工具可以为 Model 层的开发提供便利。

2. 业务逻辑层框架 Spring

Spring 是一个解决了许多 J2EE 开发中常见问题并能够替代 EJB 技术的强大的轻量级框架。这里所说的轻量级指的是 Spring 框架本身, 而不是指 Spring 只能用于轻量级的应用开发。Spring 的轻盈体现在其框架本身的基础结构以及对其他应用工具的支持和装配能力。与 EJB 这种庞然大物相比, Spring 可使程序研发人员把各个技术层次之间的风险降低。

Spring 框架的核心是控制翻转 IoC(Inversion of Control)/依赖注入 DI(Dependence Injection)机制。IoC 是指由容器中控制组件之间的关系(这里, 容器是指为组件提供特定服务和技术支持的一个标准化的运行时的环境)而非传统实现中由程序代码直接操控, 这种将控制权由程序代码到外部容器的转移, 称为“翻转”^[3]。DI 是对 IoC 更形象的解释, 即由容器在运行期间动态地将依赖关系(如构造参数、构造对象或接口)注入到组件之中^[4]。Spring 采用设值注入(使用 Setter 方法实现依赖)和构造子注入(在构造方法中实现依赖)的机制, 通过配置文件管理组建的协作对象, 创建可以构造组件的 IoC 容器。这样, 不需要编写工厂模式、单例模式或者其他构造的方法, 就可以通过容器直接获取所需的业务组件。Spring 框架的结构如图 3-2 所示。

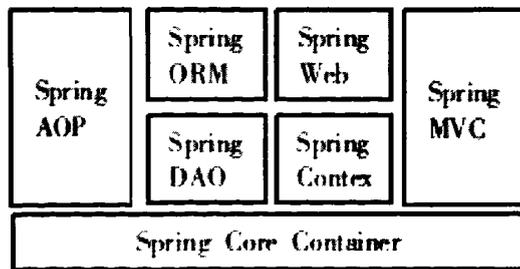


图 3-2 Spring 框架结构图

Spring 框架由七个定义明确的模块组成, 且每个模块或组件都可以单独存在, 或者与其他一个或多个模块联合实现。Spring Core Container 是一个用来管理业务组件的 IoC 容器, 是 Spring 应用的核心; Spring DAO 和 Spring ORM 不仅提供数据访问的抽象模块, 还集成了对 Hibernate、JDO 和 iBatis 等流行的对象关系映射框架的支持模块, 并且提供了缓冲连接池、事务处理等重要的服务功能, 保证了系统的性能和数据的完整性; Sprnig Web 模块提供了 Web 应用的一些抽象封装,

可以将 Struts、Webwork 等 Web 框架与 Spring 整合成为适用于自己的解决方案。

Spring 框架可以成为企业级应用程序一站式的解决方案，同时它也是模块化的框架，允许开发人员自由地挑选适合自己应用的模块进行开发。Spring 框架式是一个松耦合的框架，框架的部分耦合度被设计为最小，在各个层次上具体选用哪个框架取决于开发者的需要^[5]。

3. 数据持久层框架 Hibernate

O/R mapping 技术是为了解决关系型数据库和面向对象的程序设计之间不匹配的矛盾而产生的。Hibernate 是目前最为流行的 O/R mapping 框架，它在关系型数据库和 Java 对象之间做了一个自动映射，使得程序员可以以非常简单的方式实现对数据库的操作。Hibernate 工作原理如图 3-3 所示^[6]。

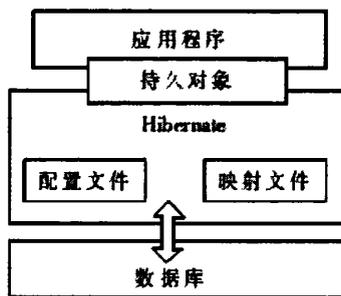


图 3-3 Hibernate 工作原理图

3.2 复合笔录技术

3.2.1 VBA word 技术

VBA 被称为宏语言，但是它决不能和宏划上等号，宏是一条指令或者几条指令的集合，控制 WORD 执行一连串的操作。而 VBA 则是不折不扣的高级语言，通过面向对象的方法来实现不能实现的功能。在编辑一个宏的时候，visual basic 编辑器会将这个宏记录为一个 VBA 过程

VBA 使应用程序具有生命的特征，以适应不同的环境，不同的应用，其主要表现为定制、自动化、协作化。

在编写程序，开发软件的过程中如果能利用已有的程序的功能，那么可以大大减轻开发过程中程序员的工作量，同时达到事半功倍的效果。例如在工程中，许多软件需要文字处理功能，虽然 MFC 提供了一些方法，但是具体实现起来既费事，又有一定的困难，要实现这一目的，只需要利用微软的 ActiveX Automation 技术就可以轻松实现^[7]。

1. 自动化的概念

在 Windows 程序开发中，自动化是基于 COM/DCOM 之上的技术：它可以使一个

应用程序来操纵另外一个程序中的对象,使用该程序提供的丰富的功能;或者是一个应用程序通过提供一些对象及对象的方法和属性来允许另外一个程序使用它提供的各种功能。其中,根据程序是受惠者还是施惠者这一原则,将提供对象和对象属性方法的应用程序称为自动化服务器端;使用对象属性方法的应用程序叫作自动化的客户端。自动化服务器让其它程序告诉它做些什么,它揭示的函数和数据称为方法(Methods)和属性(Properties)。例如 Microsoft Office 套件中的成员 Word、Excel 等都是自动化服务器。根据自动化的客户端和服务器的不同位置,又可以将自动化分为两类:

1) 如果作为服务器端和客户端的两个不同的应用程序都在同一台计算机上,叫作本地自动化;

2) 如果作为服务器端和客户端的两个不同的应用程序是基于网络的,分别处于不同的计算机上,叫作远程自动化。

需要注意的是,由于利用了 COM 技术,客户端无法直接获得服务器端的对象来实现对它的访问,它必须通过获取一个指向接口的指针来使用对象提供的功能。这个接口称为 IDispatch,它是一个简化的使用多种不同语言的特殊接口(包括象 Visual Basic 这样不能使用指针的语言)。这部分内容读者可以在下面的程序实现过程中细细体味。

说到自动化,不能不提一下 ActiveX 控件,它其实是在进程中装入的极小自动化服务器。这意味着它们的执行速度极快,它们原来被称为 OLE 自定义控件,用来替代 VB 和 Visual C++中使用的 16 位的 VBX 控件。由于 ActiveX 控件通常被保存为 .OCX 文件,所以又称它为 OCX 控件。因为控件是一个小型化的自动化服务器,它们需要在自动化客户端使用,我们又将这时的客户端应用程序称为容器应用程序。ActiveX 控件除了属性和方法外,还包含事件(events),当一些事情需要容器应用程序注意时,控件才会触发一个事件,如用户的单击等。

2. 自动化中的数据类型^[6]

在实现自动化编程中,最重要的一个问题是程序员要明白在自动化的客户端和服务端数据是如何传递的。如何提供一个统一安全的机制来实现数据传递呢?在 Visual C++开发平台上,它提供了一个叫 VARIANT 的数据类型来解决这个问题。该数据类型有两部分组成,第一部分为数据的类型,第二部分才是具体的数据的数值。在 VARIANT 数据类型的基础上,VC 提供了 ColeVariant 类,它对 VARIANT 数据类型进行了封装,这意味着在所有使用 VARIANT 的地方都可以使用 ColeVariant 类的对象。在操作日期和货币变量时,Visual C++又提供了两个类: ColeCurrency 和 ColeDateTime。

3.2.2 COM 技术

COM 是微软公司为了计算机工业的软件生产更加符合人类的行为方式开发的一种新的软件开发技术。在 COM 构架下,人们可以开发出各种各样的功能专一的组件,然后将它们按照需要组合起来,构成复杂的应用系统。由此带来的好处是多方面的:可以将系统中的组件用新的替换掉,以便随时进行系统的升级和定制;可以在多个应用系统中重复利用同一个组件;可以方便的将应用系统扩展到网络环境下;COM 与语言,平台无关的特性使所有的程序员均可充分发挥自己的才智与专长编写组件模块;等等。

COM 是开发软件组件的一种方法。组件实际上是一些小的二进制可执行程序,它们可以给应用程序,操作系统以及其他组件提供服务。开发自定义的 COM 组件就如同开发动态的,面向对象的 API。多个 COM 对象可以连接起来形成应用程序或组件系统。并且组件可以在运行时刻,在不被重新链接或编译应用程序的情况下被卸下或替换掉。Microsoft 的许多技术,如 ActiveX, DirectX 以及 OLE 等都是基于 COM 而建立起来的。并且 Microsoft 的开发人员也大量使用 COM 组件来定制他们的应用程序及操作系统。

COM 所含的概念并不止是在 Microsoft Windows 操作系统下才有效。COM 并不是一个大的 API,它实际上象结构化编程及面向对象编程方法那样,也是一种编程方法。在任何一种操作系统中,开发人员均可以遵循“COM 方法”。

一个应用程序通常是由单个的二进制文件组成的。当编译器生成应用程序之后,在对下一个版本重新编译并发行新生成的版本之前,应用程序一般不会发生任何变化。操作系统,硬件及客户需求的改变都必须等到整个应用程序被重新生成。

目前这种状况已经发生变化。开发人员开始将单个的应用程序分隔成单独多个独立的部分,也既组件。这种做法的好处是可以随着技术的不断发展而用新的组件取代已有的组件。此时的应用程序可以随新组件不断取代旧的组件而渐趋完善。而且利用已有的组件,用户还可以快速的建立全新的应用。

传统的做法是将应用程序分割成文件,模块或类,然后将它们编译并链接成一个单模应用程序。它与组件建立应用程序的过程(称为组件构架)有很大的不同。一个组件同一个小应用程序类似,即都是已经编译链接好并可以使用的二进制代码,应用程序就是由多个这样的组件打包而得到的。单模应用程序只有一个二进制代码模块。自定义组件可以在运行时刻同其他的组件连接起来以构成某个应用程序。在需要对应用程序进行修改或改进时,只需要将构成此应用程序的组件中的某个用新的版本替换掉即可。

COM,即组件对象模型,是关于如何建立组件以及如何通过组件建立应用程序的一个规范,说明了如何可动态交替更新组件。

使用组件的优点：

组件架构的一个优点就是应用可以随时间的流逝而发展进化。除此之外，使用组件还有一些可以使对已有应用的升级更加方便和灵活的优点，如应用的定制，组件库以及分布式组件等。

使用组件的种种优点直接来源于可以将它们动态的插入或卸出应用。为了实现这种功能，所有的组件必须满足两个条件：第一，组件必须动态链接；第二，它们必须隐藏（或封装）其内部实现细节。动态链接对于组件而言是一个至关重要的要求，而消息隐藏则是动态链接的一个必要条件。

3.2.3 HOOK 函数

钩子函数是系统消息处理机制的一个入口点，通过它应用程序可以安装一个例程，来监视系统的消息传递路线，对某种特定的消息在传递到目标窗口之前进行响应^[9]。

钩子没有系统的中断功能那么强大，并不能够随心所欲的截获系统的底层功能。可见，钩子是 Windows 消息机制中设置的一个监视点，应用程序可以在这里安装一个监视函数，这样就可以捕捉自己进程或者其它进程发生的事件。我们通过调用 API 函数 SetWindowsHookEx 函数就可以做到。SetWindowsHookEx 函数定义了监视函数的位置、监视消息的类型、钩子的作用范围。每当出现钩子感兴趣的消息时，Windows 就会将消息发送给监视函数。其中，监视函数由用户自己定义，是处理消息的回调函数。根据钩子处理消息的作用范围不同，Windows 所提供给我们的钩子可以分为两种类型：一是局部钩子，二是远程钩子^[10]。

局部钩子仅能够监控属于自身的事件，而远程钩子不仅可以监控自己进程中的事件，还可以用来钩挂其它进程中发生的事件。另外，远程钩子也有两种类型：其一是基于线程的，其二是基于系统的。基于线程的远程钩子是为了捕捉其它进程中某一特定线程的事件而设计的，而系统范围的远程钩子将捕捉系统中所有进程中发生的事件消息。钩子一旦安装在系统中，会影响系统的性能，因为系统发出的这些被钩子监控的事件，都要经过钩子函数的处理，特别是对于系统范围的全局钩子。所以，钩子函数中的代码要尽可能的节俭和高效，因为如果处理代码过多或者不够高效的话，系统的运行速度会受到明显的影响，这样就很容易被发现，所以对于系统范围的全局钩子一定要谨慎使用，而且，一旦钩子不用就要立即卸载掉^[11]。

钩子必备函数

要安装钩子，首先要使用 SetWindowsHookEx 函数，这个函数的原型如下：

```
HHOOK SetWindowsHookEx(  
int idHook, // 要安装的钩子的类型  
HOOKPROC lpfn, // 钩子函数的入口地址
```

```
HINSTANCE hMod, // 调用钩子函数的应用程序的实例句柄  
DWORD dwThreadId // 要在其上安装钩子的线程的 ID  
);
```

idHook 参数指定钩子的类型，钩子的类型如下表所示：

WH_CALLWNDPROC 每当调用 SendMessage 函数时，函数将消息发送给目标窗口过程前首先调用钩子函数

WH_CALLWNDPROCRET 每当调用 SendMessage 函数时，函数将消息发送给目标窗口过程后再调用钩子函数

WH_GETMESSAGE 每当调用 GetMessage 或 PeekMessage 函数时，函数从程序的消息队列中获取一个消息后调用该钩子函数

WH_KEYBOARD 每当调用 GetMessage 或 PeekMessage 函数时，如果从消息队列中得到的是 WM_KEYUP 或 WM_KEYDOWN 消息，则调用钩子函数

WH_MOUSE 每当调用 GetMessage 或 PeekMessage 函数时，如果从消息队列中得到的是鼠标消息，则调用钩子函数

WH_HARDWARE 每当调用 GetMessage 或 PeekMessage 函数时，如果从消息队列中得到的是非鼠标和键盘消息，则调用钩子函数

WH_MSGFILTER 当用户对对话框、菜单和滚动条有所操作时，系统在发送对应的消息之前调用钩子函数，这种钩子只能是局部的

WH_SYSMSGFILTER 同上，不过是系统范围的

WH_SHELL 当 Windows shell 程序准备接收一些通知事件前调用钩子函数，如 shell 被激活

WH_DEBUG 用来给其它钩子函数除错

WH_CBT 当基于计算机的训练事件发生时调用钩子函数

WH_JOURNALRECORD 日志记录钩子，用来记录发送给系统消息队列的所有消息，只能用作全局钩子

WH_JOURNALPLAYBACK 日志回放钩子，用来回放日志记录钩子记录的事件，只能用作全局钩子

WH_FOREGROUNDIDLE 系统空闲钩子，当系统空闲的时候调用钩子函数，这样就可以在这里安排一些优先级很低的任务

Lpfn 参数用来传入钩子函数的入口地址。

hInstance 参数用来指定钩子回调函数所在 DLL 的实例句柄。如果安装的是局部钩子的话，由于局部钩子的回调函数不需要放在动态链接库中，这时这个参数就使用 NULL。

DwThreadId 是安装钩子后想监控的线程的 ID 号。这个参数可以决定钩子是局部的还是系统范围的。如果参数指定的是自己进程中的某个线程的 ID 号，那么该钩子

是一个局部钩子。如果指定的线程 ID 是另一个进程中某个线程的 ID 号，那么这个钩子就是一个局部的远程钩子。如果想要安装系统范围的全局钩子的话，可以将这个参数指定为 NULL，这样钩子就会被解释成系统范围的，可以用来监控所有的进程及它们的线程。

返回值：如果钩子安装成功，函数返回钩子句柄，否则返回 NULL。需要指出的是，钩子句柄必须保存下来，因为在回调函数和卸载钩子的时候还要用到这个句柄^[12]^[13]。

卸载钩子使用函数 UnhookWindowsHookEx，这个函数的原型如下^[14]：

```
BOOL UnhookWindowsHookEx(
HHOOK hhk // 要卸载的钩子句柄
);
```

3.2.4 Activex 控件

ActiveX 是 Microsoft 对于一系列策略性面向对象程序技术和工具的称呼，其中主要的技术是组件对象模型 (COM)。在有目录和其它支持的网络中，COM 变成了分布式 COM (DCOM)。在创建包括 ActiveX 程序时，主要的工作就是组件，一个可以自足的在 ActiveX 网络（现在的网络主要包括 Windows 和 Mac）中任意运行的程序。这个组件就是 ActiveX 近控件。ActiveX 是 Microsoft 为抗衡 Sun Microsystems 的 JAVA 技术而提出的，此控件的功能和 JAVA applet 功能类似^[15]。

组件的一大优点就是可以被大多数应用程序再使用（这些应用程序称为组件容器）。一个 COM 组件（ActiveX 控件）可由不同语言的开发工具开发，包括 C++ 和 Visual Basic 或 PowerBuilder，甚至一些技术性语言如 VBScript。

目前，ActiveX 控件在 Windows 95/NT 和 Macintosh 中运行，Microsoft 还准备支持 UNIX 的 ActiveX 控件

ActiveX 组件包括如下几类：

1. 自动化服务器：可以由其他应用程序编程驱动的组件。自动化服务器至少包括一个，也许是多个供其他应用程序生成和连接的基于 IDispatch 的接口。自动化服务器可以含有也可以没有用户界面 (UI)，这取决于服务器的特性和功能。

2. 自动化控制器：那些使用和操纵自动化服务器的应用程序。

3. 控件：ActiveX 控件等价于以前的 OLE 控件或 OCX。一个典型的控件包括设计时和运行时的用户界面，唯一的 IDispatch 接口定义控件的方法和属性，唯一的 IConnectionPoint 接口用于控件可引发的事件。

4. 文档: ActiveX 文档, 即以前所说的 DocObject, 表示一种不仅仅是简单控件或自动化服务器的对象。ActiveX 文档在结构上是对 OLE 链接和模型的扩展, 并对其所在的容器具有更多控制权。一个最显著的变化是菜单的显示方式。一个典型的 OLE 文档的菜单会与容器菜单合并成一个新的集合, 而 ActiveX 文档将替换整个菜单系统, 只表现出文档的特性而不是文档与容器共同的特性^[6]。

5. 容器: ActiveX 容器是一个可以作为自动化服务器、控件和文档宿主的应用程序

ActiveX 控件是一种可重用的软件组件, 通过使用 ActiveX 控件, 可以很快地在网址、台式应用程序、以及开发工具中加入特殊的功能。如, StockTicker 控件可以用来在网页上即时地加入活动信息, 动画控件用来向网页中加入动画特性。现在, 已有 1000 多个商用的 ActiveX 控件, 开发控件可以使用各种编程语言, 如 C, C++, 下一代的 Microsoft Visual Basic[®], 以及微软公司的 Visual Java 开发环境 Microsoft Visual J++[™]。ActiveX 控件一旦被开发出来, 设计和开发人员就可以把它当作预装配组件, 用于开发客户程序。以此种方式使用 ActiveX 控件, 使用者无需知道这些组件是如何开发的, 在很多情况下, 甚至不需要自己编程, 就可以完成网页或应用程序的设计。

3.3 视频点播直播技术

3.3.1 视频点播(VOD)简介

作为一个 VOD 系统主要由三部分构成:

1) 服务端系统

服务端系统主要由视频服务器、档案管理服务器、内部通讯子系统和网络接口组成。档案管理服务器主要承担用户信息管理、计费、影视材料的整理和安全保密等任务。内部通讯子系统主要完成服务器间信息的传递、后台影视材料和数据的交换。网络接口主要实现与外部网络的数据交换和提供用户访问的接口。视频服务器主要由存储设备、高速缓存和控制管理单元组成, 其目标是实现对媒体数据的压缩和存储, 以及按请求进行媒体信息的检索和传输。视频服务器与传统的数据服务器有许多显著的不同, 需要增加许多专用的软硬件功能设备, 以支持该业务的特殊需求。例如: 媒体数据检索、信息流的实时传输以及信息的加密和解密等。对于交互式的 VOD 系统来说, 服务端系统还需要实现对用户实时请求的处理、访问许可控制、VCR (Video Cassette Recorder) 功能 (如, 快进、暂停、重绕等) 的模拟。

2) 网络系统

网络系统包括主干网络和本地网络两部分。因为它负责视频信息流的实时传输, 所以是影响连续媒体网络服务系统性能极为关键的部分。同时, 媒体服务系统

的网络部分投资巨大，故而在设计时不仅要考虑当前的媒体应用对高带宽的需求，而且还要考虑将来发展的需要和向后的兼容性。当前，可用于建立这种服务系统的网络物理介质主要是：CATV（有线电视）的同轴电缆、光纤和双绞线。而采用的网络技术主要是：快速以太网、FDDI 和 ATM 技术。

3) 客户端系统

目前，根据不同的功能需求和应用场景，主要有三种 VOD 系统：NVOD，TVOD，IVOD。

NVOD (Near-Video-On-Demand)，可称其为就近式点播电视。这种点播电视的方式是：多个视频流依次间隔一定的时间启动发送同样的内容。比如，十二个视频流每隔十分钟启动一个发送同样的两小时的电视节目。如果用户想看这个电视节目可能需要等待，但最长不会超过十分钟，他们会选择距他们最近的某个时间起点点进行收看。在这种方式下，一个视频流可能为许多用户共享。

TVOD (True Video-On-Demand)，称其为真实点播电视，它真正支持即点即放。当用户提出请求时，视频服务器将会立即传送用户所要的视频内容。若有另一个用户提出同样的需求，视频服务器就会立即为他再启动另一个传输同样内容的视频流。不过，一旦视频流开始播放，就要连续不断的播放下去，直到结束。这种方式下，每个视频流转为某个用户服务。

IVOD (Interactive Video-On-Demand)，称为交互式点播电视。它比前两种方式有很大程度上的改进。它不仅可以支持即点即放，而且还可以让用户对视频流进行交互式的控制。这时，用户就可象操作传统的录像机一样，实现节目的播放、暂停、倒回、快进和自动搜索等。

只有使用相应的终端设备，用户才能与某种服务或服务提供者进行联系和互操作。在 VOD 系统中，需要电视机和机顶盒 (Set-top Box)，在一些特殊系统中，可能还需要一台配有大容量硬盘的计算机以存储来自视频服务器的影视文件。客户端系统中，除了涉及相应的硬件设备，还需要配备相关的软件。例如，为了满足用户的多媒体交互需求，必须对客户端系统的界面加以改造。此外，在进行连续媒体播放时，媒体流的缓冲管理、声频与视频数据的同步、网络中断与演播中断的协调等问题都需要进行充分的考虑

3.3.2 典型的视频点播系统

随着时代的进步，人们对视频节目服务提出了更高的要求，希望改传统的被动观看为主动控制，能实现对视频节目的内容、播放时间和播放进度等诸多方面进行选择和控制。视频点播就是为了满足人们这种需求而产生的一种服务方式。近年来，随着高速计算机、数字视频压缩以及大容量存储器的发展，视频点播 (Video-on-Demand) 服务成为可能。一个典型的视频点播系统是由视频服务器、

高速传输网络和客户端组成，服务器响应用户的点播请求，将节目数据以视频流的形式通过高速传输网络发送到用户端。视频服务器是整个视频点播系统的核心部分，选择和安装性价比高的视频服务器是构建视频点播系统的关键。图 3-4 所示是一个典型的 VOD 的系统结构。

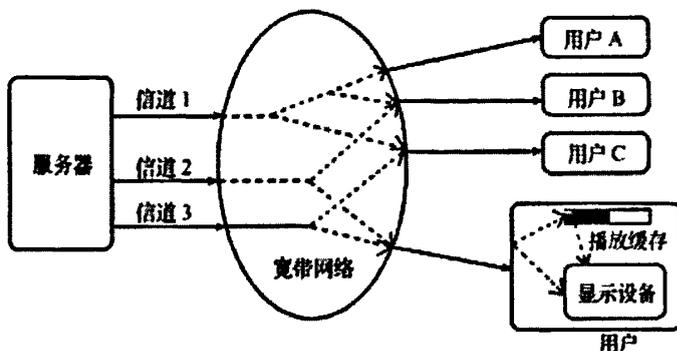


图 3-4 典型的视频点播结构图

1、VOD 服务器的选择

目前市场上有三大主流 VOD 服务器，即 RealServer、Windows Media Services 和 QuickTime。

(1) RealServer

REAL 公司是世界上最早开发流格式的公司，是世界领先的网上流式技术解决方案的提供者。最新的市场调研结果显示，REAL 公司仍然牢牢控制着流式媒体市场。RealServer 目前有 4 个版本：Basi、Plus、Professional 和 Intranet。RealServer 是 3 种主要 VOD 服务器中惟一收费的(Basic 版本是免费的但是只能同时支持 25 个流)。另外，它还支持 MP3 和 QuickTime 文件作为流发送。

(2) Windows Media Services

Windows Media Services (WMS) 只能运行在基于 Windows 的服务器上，Windows 2000 Server 和 Advanced Server 都免费附加了 WMS。MMS 使用自己的 MMS 协议支持高级流格式 (ASF) 文件。ASF 文件可以有一些其他的文件扩展名，如 .wma 和 .wmv 等。

(3) QuickTime Streaming Server

QuickTime 公司的 VOD 服务器只能运行于 Mac 的硬件上，它支持 QuickTime 格式的流媒体文件，即以 .mov 为扩展名的文件。

2、选择合适的 VOD 服务器

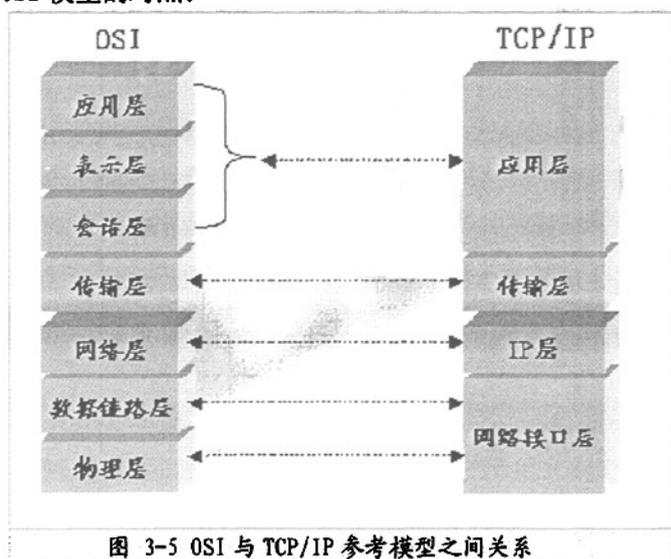
对于许多流媒体提供者来说，操作系统是决定因素。如果使用的是 UNIX，那么应该选择 RealServer，因为 WMS 不能在 UNIX 上运行。如果使用的是 Windows2000 Server，因为 WMS 和操作系统结合得非常紧密，并且包含于操作系统中不需付费，所以应该选择 WMS。如果使用的是 Mac OS，那么 QuickTime Streaming Server 将是惟一的选择。需要注意的是，所选用的 VOD 服务器决定了其所支持的流媒体文件类型。

3.3.3 视频点播的相关协议

1) TCP/IP 协议

TCP/IP 出现于 20 世纪 70 年代，80 年代被确定为因特网的通信协议。

TCP/IP 参考模型是将多个网络进行无缝连接的体系结构，其模型如下图 3-5 所示，中加入了与 OSI 模型的对照。



TCP/IP 是一组通信协议的代名词，是由一系列协议组成的协议簇。它本身指两个协议集：TCP(传输控制协议)和 IP(互联网络协议)的结合而成的。

下面简单地介绍 TCP/IP 协议的各层所提供的服务。

1) 主机到网络层(网络接口层)

TCP/IP 参考模型对 IP 层之下未加定义，只指出主机必须通过某种协议连接到网络，才能发送 IP 分组。该层协议未定义，随不同主机、不同网络而不同，因此主机到网络层又称为网络接口层。

这是 TCP/IP 模型的最低层，负责接收从 IP 层交来的 IP 数据报并将 IP 数据报通过低层物理网络发送出去，或者从低层物理网络上接收物理帧，抽出 IP 数据报，交给 IP 层。网络接口有两种类型。第一种是设备驱动程序，如局域网的网络接口；第二种是含自身数据链路协议的复杂子系统。TCP/IP 未定义数据链路层，是因为在 TCP/IP 最初的设计中已经使其可以使用包括以太网、令牌环网、FDDI 网、ISDN 和 X.25 在内的多种数据链路层协议。

TCP/IP 可使用于多种传输介质。例如，在以太网中，TCP/IP 可支持同轴电缆、双绞线和光纤。TCP/IP 在 x.25 上的应用可以支持微波传输或电话线路。

2) 互连网络层(IP 层)

互联网层的主要功能是负责相邻结点之间的数据传送。它的主要功能包括三个方面。第一，处理来自传输层的分组发送请求：将分组装入 IP 数据报，填充报头，

选择去往目的结点的路径，然后将数据报发往适当的网络接口。第二，处理输入数据报：首先检查数据报的合法性，然后进行路由选择，假如该数据报已到达目的结点（本机），则去掉报头，将 IP 报文的数据部分交给相应的传输层协议；假如该数据报尚未到达目的结点，则转发该数据报。第三，处理 ICMP 报文：即处理网络的路由选择、流量控制和拥塞控制等问题。TCP/IP 网络模型的互联网层在功能上非常类似于 OSI 参考模型中的网络层。

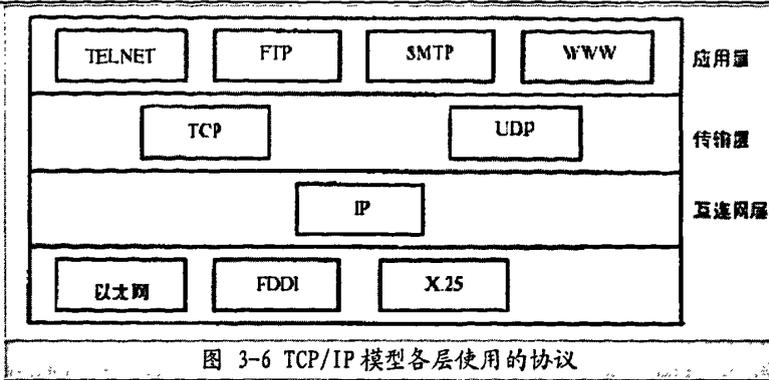
互连网络层是网络互连的基础，提供了无连接的分组交换服务，它是对大多数分组交换网所提供服务的抽象。其任务是允许主机将分组放到网上，让每个分组独立地到达目的地。分组到达的顺序可能不同于分组发送的顺序，由高层协议负责对分组重新进行排序。与避免拥挤一样，分组的路径选择是本层的主要工作。

由于在 IP 层提供数据报服务，常将报文分组称为 IP 数据报

3) 传输层(TCP)

TCP/IP 参考模型中传输层的作用与 OSI 参考模型中传输层的作用是一样的，即在源结点和目的结点的两个进程实体之间提供可靠的端到端的数据传输。为保证数据传输的可靠性，传输层协议规定接收端必须发回确认，并且假定分组丢失，必须重新发送。传输层还要解决不同应用程序的标识问题，为在一般的通用计算机中，常常是多个应用程序同时访问互联网。为区别各个应用程序，传输层在每一个分组中增加识别信源和信宿应用程序的标记。另外，传输层的每一个分组均附带校验和，以便接收结点检查接收到的分组的正确性。

TCP/IP 模型提供了两个传输层协议：传输控制协议 TCP 和用户数据报协议 UDP。TCP 协议是一个可靠的面向连接的传输层协议，它将某结点的数据以字节流形式无差错投递到互联网的任一台机器上。发送方的 TCP 将用户交来的字节流划分成独立的报文并交给互联网层进行发送，而接收方的 TCP 将接收的报文重新装配交给接收用户。TCP 同时处理有关流量控制的问题，以防止快速的发送方淹没慢速的接收方。用户数据报协议 UDP 是一个不可靠的、无连接的传输层协议，UDP 协议将可靠性问题交给应用程序解决。UDP 协议主要面向请求/应答式的交易型应用，一次交易往往只有一来一回两次报文交换，假如为此而建立连接和撤销连接，开销是相当大的。这种情况下使用 UDP 就非常有效。另外，UDP 协议也应用于那些对可靠性要求不高，但要求网络的延迟较小的场合，如话音和视频数据的传送。IP、TCP 和 UDP 的关系如图 3-6 所示。



4) 应用层

TCP/IP 参考模型中没有会话层与表示层。OSI 模型的实践发现，大部分的应用程序不涉及这两层，故 TCP/IP 参考模型不予考虑。在传输层之上就是应用层，它包含了所有高层协议。早期高层协议有虚拟终端协议 (Telnet)，文件传输协议 (FTP)，电子邮件传输协议 (SMTP)。

如上图所示，Telnet 允许用户登录到远程机器并在其上工作。文件传输协议 TFTP 提供了有效地将数据从一台机器传送到另一台机器的机制。早期的电子邮件仅仅是文件传送，后来为它开发了专门的协议 SMTP。在应用层又加入了许多其他协议，如域名服务 (DNS) 用于将主机名映射到它们的网络地址，NNTP 是用于发布新闻的协议，HTTP 是用于搜索 WWW 上超文本的协议等等。

5) TCP/IP 的基本工作原理

从以上体系结构来看，TCP/IP 是 OSI 七层模型的简化，如图 3-7 所示，共分为 4 层：应用层，传输层，IP 层和物理网络接口层。TCP/IP 模型将与物理网络打交道的物理网络部分称为网络接口，它相当于 OSI 的物理层和数据链路层。

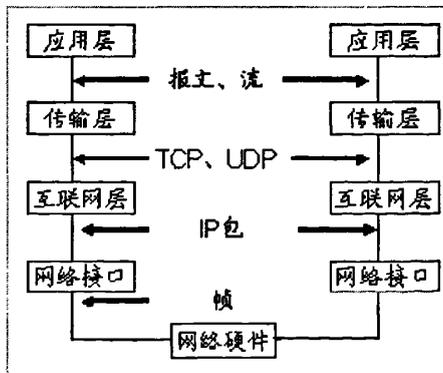


图 3-7 TCP/IP 是 OSI 七层模型的简化

在下表中给出了 TCP/IP 协议集的主要协议及它们所提供的服务,如表 3-1 所示。

表 3-1 协议服务及其提供服务

协议	提供服务	相应OSI
IP	数据报服务	3
ICMP	差错和控制	3
ARP	互联网地址—>物理地址	3
RARP	物理地址—>互联网地址	3
TCP	可靠流服务	4
FTP	文件传送	7
TELNET	终端仿真	7
DNS	域名—IP地址	7

上图 3-7 所示,在互联网上源主机的协议层与目的主机的同层协议通过下层提供的服务实现对话。在源和目的主机的同层实体称为对等实体(Peer entities)或叫对等进程,它们之间的对话实际上是在源主机上从上到下然后穿越网络到达目的主机后再从下到上到达相应层。

下面以使用 TCP 协议传送文件(如 FTP 应用程序)为例说明了 TCP/IP 的工作原理。

- (1) 在源主机上应用层将一串字节流传给传输层;
 - (2) 传输层将字节流分成 TCP 段,加上 TCP 包头交给互联网络(IP)层;
 - (3) IP 层生成一个包,将 TCP 段放入其数据域,并加上源和目的主机的 IP 包交给数据链路层;
 - (4) 数据链路层在其帧的数据部分装 IP 包,发往日的主机或 IP 路由器;
 - (5) 在目的主机,数据链路层将数据链路层帧头去掉,将 IP 包交给互联网层;
 - (6) IP 层检查 IP 包头,如果包头中的校验和与计算出来的不一致,则丢弃该包;
 - (7) 如果校验和一致,IP 层去掉 IP 头,将 TCP 段交给 TCP 层,TCP 层检查顺序号来判断是否为正确的 TCP 段;
 - (8) TCP 层为 TCP 包头计算 TCP 头和数据。如果不对,TCP 层丢弃这个包,若对,则向源主机发送确认;
 - (9) 在目的主机,TCP 层去掉 TCP 头,将字节流传给应用程序;
 - (10) 于是目的主机收到了源主机发来的字节流,就像直接从源主机发来的一样。实际上每往下一层,便多加了一个报头,而这个头对上层来说是透明的,上层根本感觉不到下面报头的存在。
- 2) RTP 协议

RTP 是一种提供端对端传输服务的实时传输协议,用来支持在单目标广播和多目标广播网络服务中传输实时数据,而实时数据的传输则由 RTCP 协议来监视和控制。RTP 定义在 RFC 使用 RTP 协议的应用程序运行在 RTP 之上,而执行 RTP 的程序运行在 UDP 的上层,目的是为了使用 UDP 的端口号和检查。如图 3-8 所示,RTP 可以看成是传输层的子层。由多媒体应用程序生成的声音和电视数据块被封装在 RTP 信息包中,每个 RTP 信息包被封装在 UDP 消息段中,然后再封装在 IP 数据包中^[17]。

	TCP/IP 模型
	应用层(application)
传输层	RTP
	UDP
	IP
	数据链路层(data link)
	物理层(physical)

图 3-8 RTP 是传输层上的协议

从应用开发人员的角度来看,可把 RTP 执行程序看成是应用程序的一部分,因为开发人员必需把 RTP 集成到应用程序中。在发送端,开发人员必需把执行 RTP 协议的程序写入到创建 RTP 信息包的应用程序中,然后应用程序把 RTP 信息包发送到 UDP 的套接接口(socket interface),如图 3-9 所示;同样,在接收端,RTP 信息包通过 UDP 套接接口输入到应用程序,因此开发人员必需把执行 RTP 协议的程序写入到从 RTP 信息包中抽出媒体数据的应用程序。

TCP/IP 模型	
应用层(application)	
RTP	
	套接接口
UDP	
IP	
数据链路层(data link)	
物理层(physical)	

图 3-9 RTP 和 UDP 之间的接口

现以用 RTP 传输声音为例来说明它的工作过程。假设音源的声音是 64 kb/s 的 PCM 编码声音,并假设应用程序取 20 毫秒的编码数据为一个数据块(chunk),即在一个数据块中有 160 个字节的声数据。应用程序需要为这块声音数据添加 RTP 标题生成 RTP 信息包,这个标题包括声音数据的类型、顺序号和时间戳。然后 RTP 信息包被送到 UDP 套接接口,在那里再被封装在 UDP 信息包中。在接收端,应用程序从套接接口处接收 RTP 信息包,并从 RTP 信息包中抽出声音数据块,然后使用 RTP 信息包的标题域中的信息正确地译码和播放声音。

如果应用程序不使用专有的方案来提供有效载荷类型(payload type)、顺序号或者时间戳,而是使用标准的 RTP 协议,应用程序就更容易与其他的网络应用程序配合运行,这是大家都希望的事情。例如,如果有两个不同的公司都在开发因特网

电话软件，他们都把 RTP 合并到他们的产品中，这样就有希望：使用不同公司电话软件的用户之间能够进行通信。

这里需要强调的是，RTP 本身不提供任何机制来确保把数据及时递送到接收端或者确保其他的服务质量，它也不担保在递送过程中不丢失信息包或者防止信息包的次序不被打乱。的确，RTP 的封装只是在系统端才能看到，中间的路由器并不区分那个 IP 数据报是运载 RTP 信息包的。

RTP 允许给每个媒体源分配一个单独的 RTP 信息包流，例如，摄像机或者麦克风。例如，有两个团体参与的电视会议，这就可能打开 4 个信息包流：两台摄像机传送电视流和两个麦克风传送声音流。然而，许多流行的编码技术，包括 MPEG-1 和 MPEG-2 在编码过程中都把声音和电视图像捆绑在一起以形成单一的数据流，一个方向就生成一个 RTP 信息包流。

RTP 信息包没有被限制只可应用于单目标广播，它们也可以在一对多 (one-to-many) 的多目标广播树或者在多对多 (many-to-many) 的多目标广播树上传送。例如，多对多的多目标广播，在这种应用场合下，所有发送端通常都把他们的 RTP 信息包流发送到具有相同多目标广播地址的多目标广播树上。

RTP 信息包标题域^[19]

RTP 标题由 4 个信息包标题域和其他域组成：有效载荷类型 (payload type) 域，序号 (sequence number) 域，时间戳 (timestamp) 域和同步源标识符 (Synchronization Source Identifier) 域等。RTP 信息包的标题域的结构如下图 3-10 所示：

Payload Type (有效载荷类型)	Sequence Number (序号)	Timestamp (时间戳)	Synchronization Source Identifier (同步源标识符)	Miscellaneous Fields (其他)
-----------------------------	-------------------------	--------------------	---	---------------------------------

图 3-10 RTP 信息包的标题域的结构图

1. 有效载荷类型

RTP 信息包中的有效载荷域 (Payload Type Field) 的长度为 7 位，因此 RTP 可支持 128 种不同的有效载荷类型。对于声音流，这个域用来指示声音使用的编码类型，例如 PCM、自适应增量调制或线性预测编码等等。如果发送端在会话或者广播的中途决定改变编码方法，发送端可通过这个域来通知接收端。表 3-2 列出了目前 RTP 所能支持的声音有效载荷类型。

表 3-2 目前 RTP 所能支持的声音有效载荷类型

有效载荷号	声音类型	采样率(kHz)	数据率(kb/s)
0	PCM mu-law	8	64
1	1016	8	4.8
2	G.721	8	32
3	GSM	8	32
6	DVI	16	64
7	LPC	8	2.4
9	G.722	8	48~64
14	MPEG Audio	90	-
15	G.728	8	16

对电视流,有效载荷类型可以用来指示电视编码的类型,例如 motion JPEG, MPEG-1, MPEG-2 或者 H.231 等等。发送端也可以在会话或者期间随时改变电视的编码方法。表 3-3 列出了目前 RTP 所能支持的某些电视有效载荷类型。

表 3-3 目前 RTP 所能支持的声音有效载荷类型

有效载荷号	电视格式
26	Motion JPEG
28	-
31	H.261
32	MPEG-1 video
33	MPEG-2 video

2. 顺序号

顺序号(Sequence Number Field)域的长度为 16 位。每发送一个 RTP 信息包顺序号就加 1,接收端可以用它来检查信息包是否有丢失以及按顺序号处理信息包。例如,接收端的应用程序接收到一个 RTP 信息包流,这个 RTP 信息包在顺序号 86 和 89 之间有一个间隔,接收端就知道信息包 87 和 88 已经丢失,并且采取措施来处理丢失的数据。

3. 时间戳

时间戳(Timestamp)域的长度为 32 字节。它反映 RTP 数据信息包中第一个字节的采样时刻(时间)。接收端可以利用这个时间戳来去除由网络引起的信息包的抖动,并且在接收端为播放提供同步功能。

4. 同步源标识符

同步源标识符(Synchronization Source Identifier, SSRC)域的长度为 32 位。它用来标识 RTP 信息包流的起源,在 RTP 会话或者期间的每个信息包流都有一个清楚的 SSRC。SSRC 不是发送端的 IP 地址,而是在新的信息包流开始时源端随机分配的一个号码。

3.3.4 SOCKET 技术

socket 实质上提供了进程通信的端点。进程通信之前,双方首先必须各自创建一个端点,否则是没有办法建立联系并相互通信的。正如打电话之前,双方必须各自拥有一台电话机一样。在网间网内部,每一个 socket 用一个半相关描述:(协议,本地地址,本地端口)。一个完整的 socket 有一个本地唯一的 socket 号,由操作系统分配。socket 是面向客户/服务器模型而设计的,针对客户和服务程序提供不同的 socket 系统调用。客户随机申请一个 socket (相当于一个想打电话的人可以在任何一台入网电话上拨号呼叫),系统为之分配一个 socket 号;服务器拥有全局公认的 socket,任何客户都可以向它发出连接请求和信息请求(相当于一个被呼叫的电话拥有一个呼叫方知道的电话号码)。

socket 利用客户/服务器模式巧妙地解决了进程之间建立通信连接的问题。服务器 socket 半相关为全局所公认非常重要。

一、客户机/服务器模式

在 TCP/IP 网络中两个进程间的相互作用的主机模式是客户机/服务器模式(Client/Server model)。该模式的建立基于以下两点:1、非对等作用;2、通信完全是异步的。客户机/服务器模式在操作过程中采取的是主动请示方式:首先服务器方要先启动,并根据请示提供相应服务:(过程如下)

- 1、打开一通信通道并告知本地主机,它愿意在某一个公认地址上接收客户请求。
- 2、等待客户请求到达该端口。
- 3、接收到重复服务请求,处理该请求并发送应答信号。
- 4、返回第二步,等待另一客户请求
- 5、关闭服务器。

客户方:

- 1、打开一通信通道,并连接到服务器所在主机的特定端口。
- 2、向服务器发送服务请求报文,等待并接收应答;继续提出请求……
- 3、请求结束后关闭通信通道并终止。

二、基本套接字

为了更好说明套接字编程原理,给出几个基本的套接字,在以后的篇幅中会给出更详细的使用说明。

1、创建套接字——socket()

功能: 使用前创建一个新的套接字

格式: SOCKET PASCAL FAR socket(int af, int type, int protocol);

参数: af: 通信发生的区域

type: 要建立的套接字类型

protocol: 使用的特定协议

2、指定本地地址——bind()

功能: 将套接字地址与所创建的套接字号联系起来。

格式: int PASCAL FAR bind(SOCKET s, const struct sockaddr FAR * name, int namelen);

参数: s: 是由 socket() 调用返回的并且未作连接的套接字描述符(套接字号)。

其它: 没有错误, bind() 返回 0, 否则 SOCKET_ERROR

地址结构说明:

```
struct sockaddr_in
```

```
{
```

```
short sin_family;//AF_INET
```

```
u_short sin_port;//16 位端口号, 网络字节顺序
```

```
struct in_addr sin_addr;//32 位 IP 地址, 网络字节顺序
```

```
char sin_zero[8];//保留
```

```
}
```

3、建立套接字连接——connect() 和 accept()

功能: 共同完成连接工作

格式: int PASCAL FAR connect(SOCKET s, const struct sockaddr FAR * name, int namelen);

SOCKET PASCAL FAR accept(SOCKET s, struct sockaddr FAR * name, int FAR * addrlen);

参数: 同上

4、监听连接——listen()

功能: 用于面向连接服务器, 表明它愿意接收连接。

格式: int PASCAL FAR listen(SOCKET s, int backlog);

5、数据传输——send() 与 recv()

功能: 数据的发送与接收

格式: int PASCAL FAR send(SOCKET s, const char FAR * buf, int len, int flags);

```
int PASCAL FAR recv(SOCKET s, const char FAR * buf, int len, int flags);
```

参数: buf:指向存有传输数据的缓冲区的指针。

6、多路复用——select()

功能: 用来检测一个或多个套接字状态。

```
格式: int PASCAL FAR select(int nfd, fd_set FAR * readfds, fd_set FAR * writefds,
```

```
fd_set FAR * exceptfds, const struct timeval FAR * timeout);
```

参数: readfds:指向要做读检测的指针

writefds:指向要做写检测的指针

exceptfds:指向要检测是否出错的指针

timeout:最大等待时间

7、关闭套接字——closesocket()

功能: 关闭套接字 s

```
格式: BOOL PASCAL FAR closesocket(SOCKET s);
```

三、典型过程图

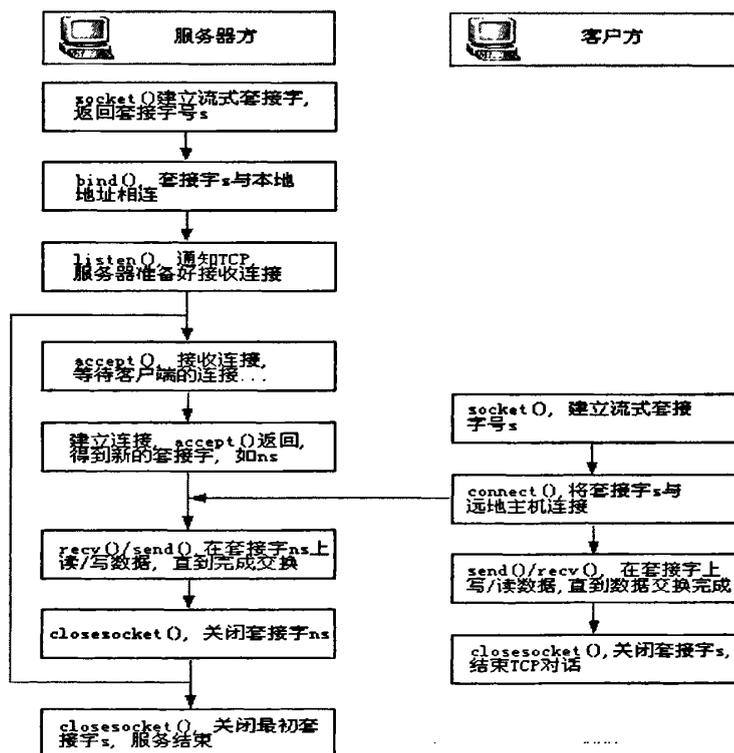


图 3-11 面向连接的套接字的系统调用时序图

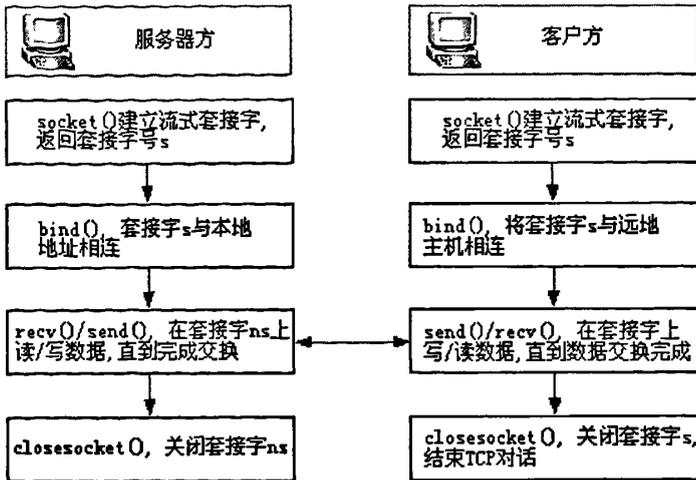


图 3-12 无连接协议的套接字调用时序图

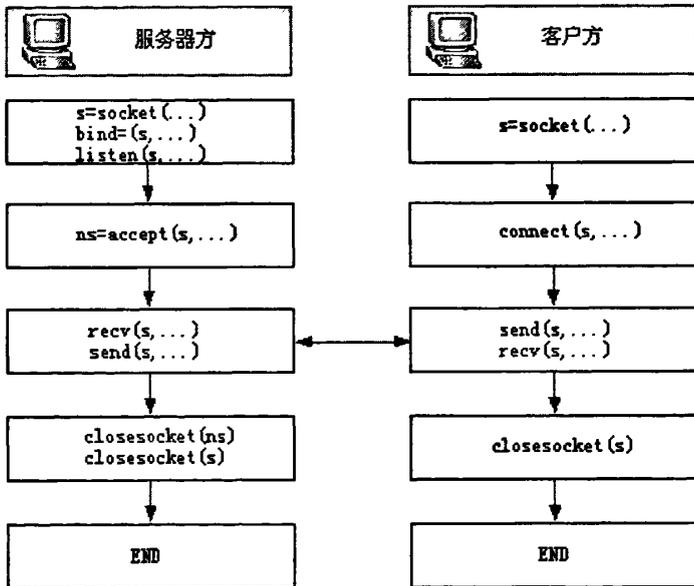


图 3-13 面向连接的应用程序流程图

3.3.5 DirectShow 技术

1) 系统构架

DirectShow 是 DirectX 大家族中的一个成员。它为 Windows 平台上处理各种各样的媒体文件播放，音视频采集等高性能要求的多媒体应用，提供了完整的解决方案^{[20] [21]}。

开发多媒体应用要面临很多问题，如：如何保证多媒体数据处理的高效性；如何让音频和视频保持同步；如何用简单的方法处理复杂的媒体源文；如何处理各

种的媒体格式问题（包括 AVI, ASF, MPEG, DV, MOV 等）；如何支持目标系统中不可预知的硬件。微软公司提出的 DirectShow，实际上就是为了解决上述这些问题的。DirectShow 设计的初衷，就是尽量要让应用程序开发者从复杂的解决数据传输、硬件差异、同步性等问题的工作中解脱出来；总体的应用框架和底层工作由 DirectShow 来完成，而让开发者把精力专注于具体的多媒体数据的处理上。DirectShow 是一个开放性的应用框架，也是一套基于 GOM 的编程接口。DirectShow 的系统功能参见图 3-13。可以看到，图中最大的一块即是 DirectShow 系统，它的基本工作原理就是“流水线”：将单元组件——Filter——串联在一起，交由 FilterGraphManager 统一控制。系统的输入可以是本地文件系统、硬件插卡、因特网等，系统的输出可以是声卡（声音再现）、显卡（视频内容显示）、本地文件系统，当然也可以最终将数据向网络发送。如图 3-14 所示，图中央最大的一块即是 DirectShow 系统，虚线以下是“Ring。”特权级别的硬件设备，虚线以上是“Ring3”特权级别的应用层。DirectShow 系统位于应用层中。它使用一种叫 FilterGraph 的模型来管理整个数据流的处理过程；参与数据处理的各个功能模块叫做 Filter；各个 Filter 在 FilterGraph 中按一定的顺序连接成一条“流水线”协同工作。按照功能来分，Filter 大致分为三类：源(Source)Filter、转换(Transform)Filter 和渲染(Rendering)Filter。源 Filter 主要负责取得数据，数据源可以是文件、因特网、或者计算机里的采集卡（WDM 驱动的或 VFW 驱动的）、数字摄像机等，然后将数据往下传输；转换 Filter 主要负责数据的格式转换，比如数据流分离/合成、解码/编码等，以及继续往下传输；渲染 Filter 主要负责数据的最终去向——将数据——送给显卡、声卡进行多媒体的演示，或者输出到文件进行存储^{[23] [24]}。

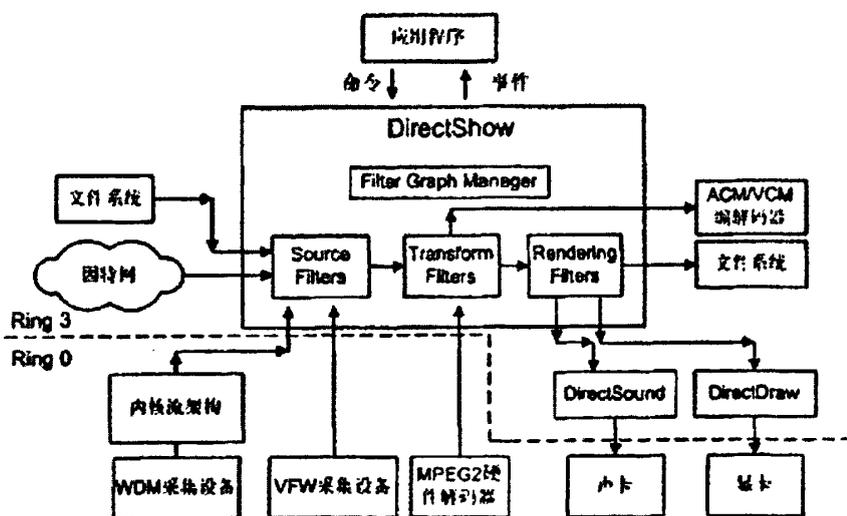


图 3-14 DirectShow 的系统图

2) Filter 与 FilterGraph

一个 DirectShow 程序是由一种叫做 Filter 的基本单元构建成的。一个 Filter

是一个软件组件,它能够完成某种对多媒体流的操作。如:读文件、编解码、输出到显卡等等。Filter 就像一个黑匣子,接受输入、产程输出,在内部对流(多媒体数据或其他形式的数据)进行特定的处理。

整个 DirectShow 程序就是把若干个 Filter 连接在一起,来完成某种功能。一组连接在一起的 Filter 称为一个 FilterGraph。例如,图 3-15 表示了一个播放 AVI 文件的 FilterGraph。

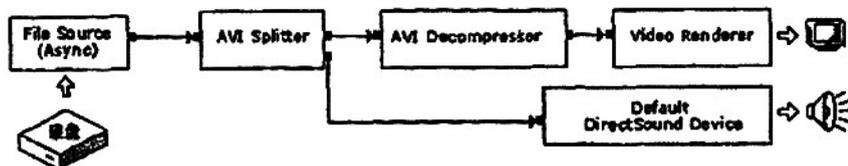


图 3-15 AvI 的 FilterGraph

3) FilterGraph Manager

应用程序不必管理所有的数据流,Filter 由一种被称为 Filter Graph Manager 得更高级的组建控制。应用程序只要调用高级的 API 即可,如调用 Run 来使数据在图表中移动,调用 Stop 来停止数据流。Filter Graph Manager 也向应用程序传递事件通知。

我们也使用 Filter Graph Manger 来组建 Filter Graph,即把各个 Filter 连接在一起,组成一个 Filter Graph^[18]

4) DirectShow 程序开发步骤

一般来说,开发一个 DirectShow 程序需要完成下面三个步骤,如图 3-16 所示:

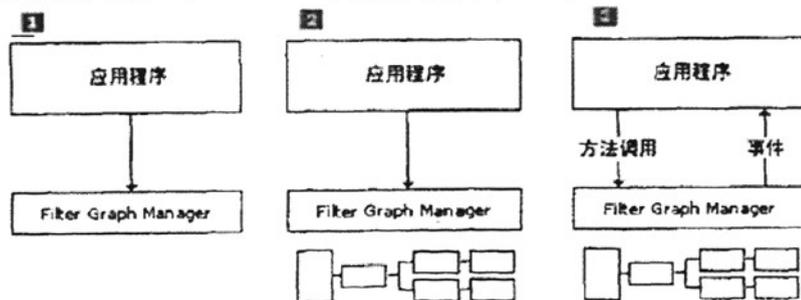


图 3-16 开发步骤图

- (1) 创建一个 Filter Graph Manager 的实例子;
- (2) 使用 Filter Graph Manager 构建一个 Filter Graph;
- (3) 使用 Filter Graph Manager 控制 Filter Graph 和通过 Filter 的流数据。在这个过程中,应用程序需要处理 Filter Graph Manger 传递出的事件。

3.4 设计模式

本项目里用到了一种设计模式叫观察者模式，又叫发布-订阅 (Publish/Subscribe) 模式，定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。这个主题对象在状态发生变化时，会通知所有观察者对象，使它们能够自动更新自己^{[25] [26]}。如表 3-4

名称	Observer (观察者模式)
意图	定义对象间的一种一对多的依赖关系,当一个对象的状态发生改变时,所有依赖于它的对象都得到通知并被自动更新。
适用环境	<ul style="list-style-type: none"> ▣ 当一个抽象模型有两个方面,其中一个方面依赖于另一方面。将这二者封装在独立的对象中以使它们可以各自独立地改变和复用。 ▣ 当对一个对象的改变需要同时改变其它对象,而不知道具体有多少对象有待改变。 ▣ 当一个对象必须通知其它对象,而它又不能假定其它对象是谁。换言之,你不希望这些对象是紧密耦合的^{[27] [28]}。

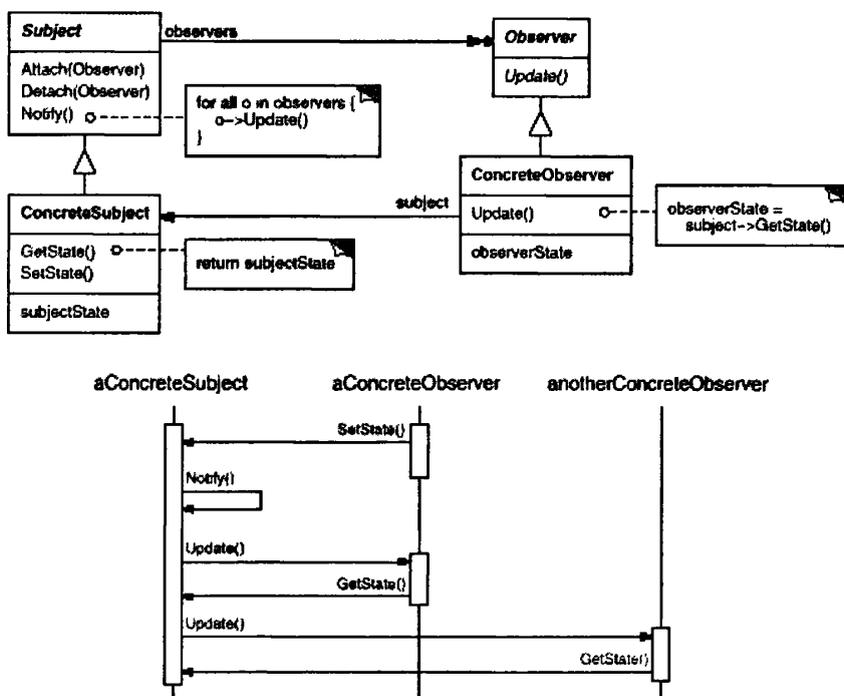


图 3-17 观察者模式流程图

3.5 本章小结

本章前三节研究了数字法庭系统相关的得 3 个方面内容：网站、复合笔录技术和视频点播直播技术，并且分别介绍了与上述三个方面相关的关键技术。第四节介

绍了一种设计模式。这种设计模式为面向对象编程提供了一种新的消息机制。在后面的系统完成和实现中使用到了这些技术和设计模式。

第四章 数字法庭系统关键技术的应用

4.1 复合笔录系统的实现

4.1.1 复合笔录的设计

复合笔录模块外部结构图如下：

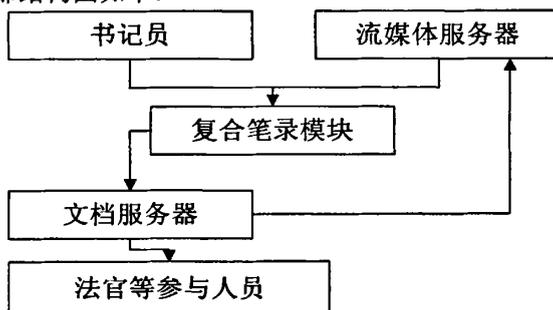


图4-1复合笔录模块外部结构图

- 书记员：负责向笔录模块录入实时文字信息
- 流媒体服务器：给笔录模块传送直播信号
- 法官等其他参与人员：观看书记员记录的文字信息
- 复合笔录模块：
 1. 接收书记员的录入
 2. 接收流媒体服务器传送过来的媒体流
 3. 发送文档流到 文档服务器上。
- 文档服务器：
 1. 接收笔录模块传送过来的实时文档流
 2. 提供文档服务，供法官等人员观看书记员所记文档

2.复合笔录模块内部结构图如下：

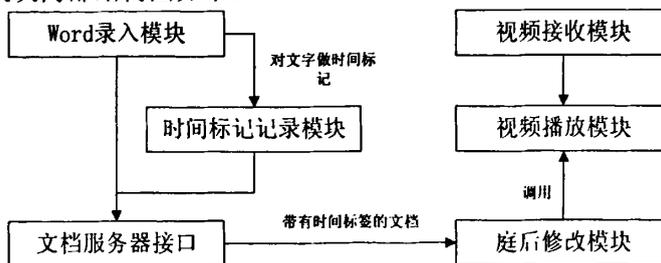


图4-2复合笔录模块内部结构图

庭审中相关模块：

- 文档录入模块：文字录入

- 视频接收模块：对外接口，接收外部视频文件
- 视频播放模块：播放所接收的视频文件
- 时间标记记录模块：记录文字对应的时间信息
- 文档服务器接口：对外接口，实时发送文字信息到外部结构设计
庭审后修改相关模块
- 庭后修改模块：接收 文档服务器传送过来的带有时间标签的文档，根据文档调用相应的视频信息。
- 视频接收模块：对外接口，接收外部视频文件
- 视频播放模块：播放所接收的视频文件

4.1.2 复合笔录的实现

1) 界面的实现

复合笔录界面部分将利用微软提供的 Activex 控件，利用 COM 技术来实现。实现的界面如图 4-3:

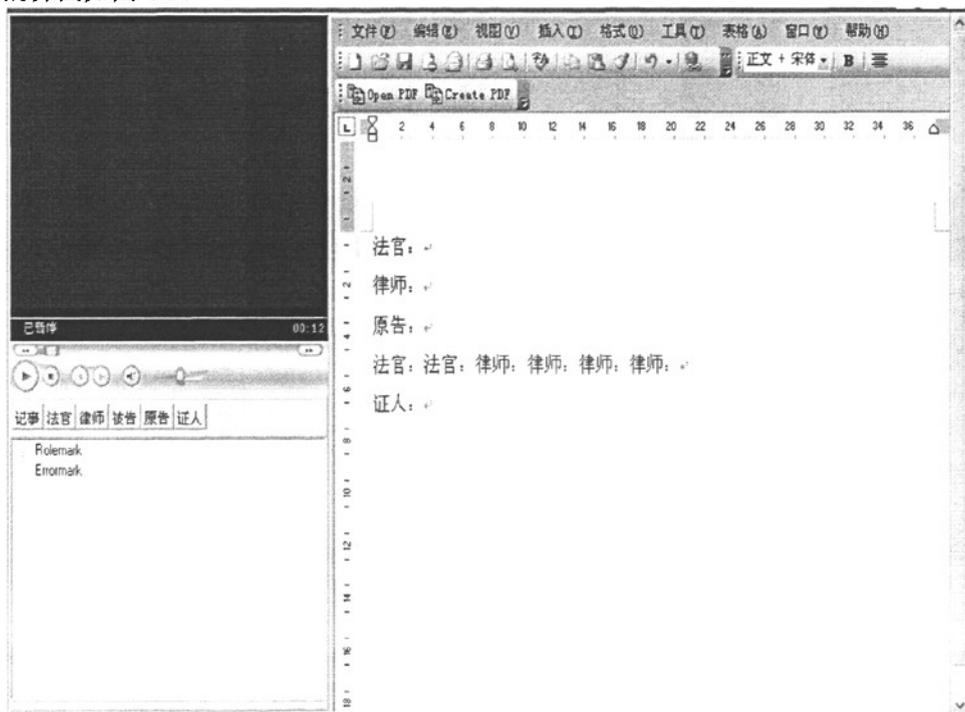


图 4-3 复合笔录界面图

2) 功能的实现

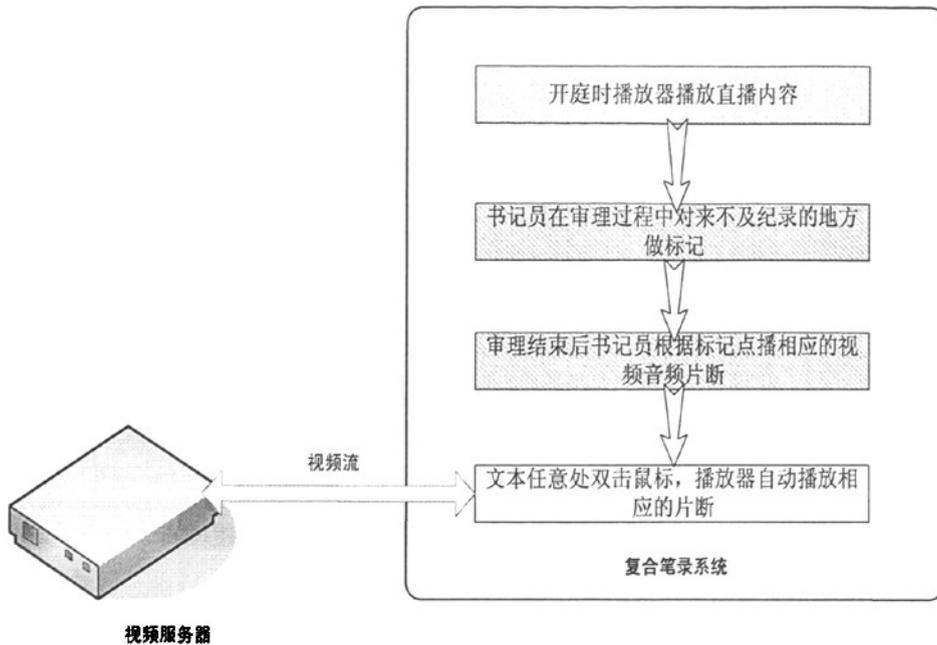


图 4-4 复合笔录流程图

这个部分要完成如下几个功能：

- 1) 开庭时播放器播放直播内容
- 2) 书记员在审理过程中对来不及纪录的地方做标记
- 3) 审理结束后书记员根据标记点播相应的视频音频片断
- 4) 审理结束后书记员在文本任意处双击鼠标，播放器自动播放相应的片断

下面分别来介绍四个功能的实现。

1) 开庭时播放器播放直播内容的实现

本项目用到音频视频采集设备，将采集到的音视频信号传输到输出端，由输出端直播的同时，将其保存为 avi 各式的文件，庭审结束后上传至媒体服务器。

2) 书记员在审理过程中对来不及纪录的地方做标记的实现

在本程序里书记员对来不及纪录的地方做标记是通过利用 Word 编程，通过打书签 Bookmark 的方式来记录。Word 文档里可以保存书签，直至人为的删除掉。

3) 审理结束后书记员根据标记点播相应的视频音频片断

书记员做的标记通过目录树来记录，根据标记调用相应的视频片断是通过对一个目录树的的点击来完成的

4) 审理结束后书记员在文本任意处双击鼠标，播放器自动播放相应的片断

这个过程是一个复杂的过程，因为要涉及到两个独立的程序，怎样通过双击 Word 文档里的任意处来调用相应的视频片断，是要用得 HOOK 函数，截取到 Word 文档里的双击事件，将双击处的位置信息传递给播放器来完成。

4.2 视频点播直播系统的实现

在本项目设计了两个方案：

1) 方案一：

利用 Windows Media Services 9

操作系统：Windows 2003 Server

必须使用其支持的格式，如 asf, wmv 等。

优点：技术成熟，便于实现

缺点：项目中使用的硬件 VideoServer 只能提供 avi 格式的文件，所以必须要转换成 asf, wmv 格式文件才可以使用，但是转换过程很慢。

2) 方案二：

利用 DirectShow 技术结合 RTP 协议做简单的视频服务器。

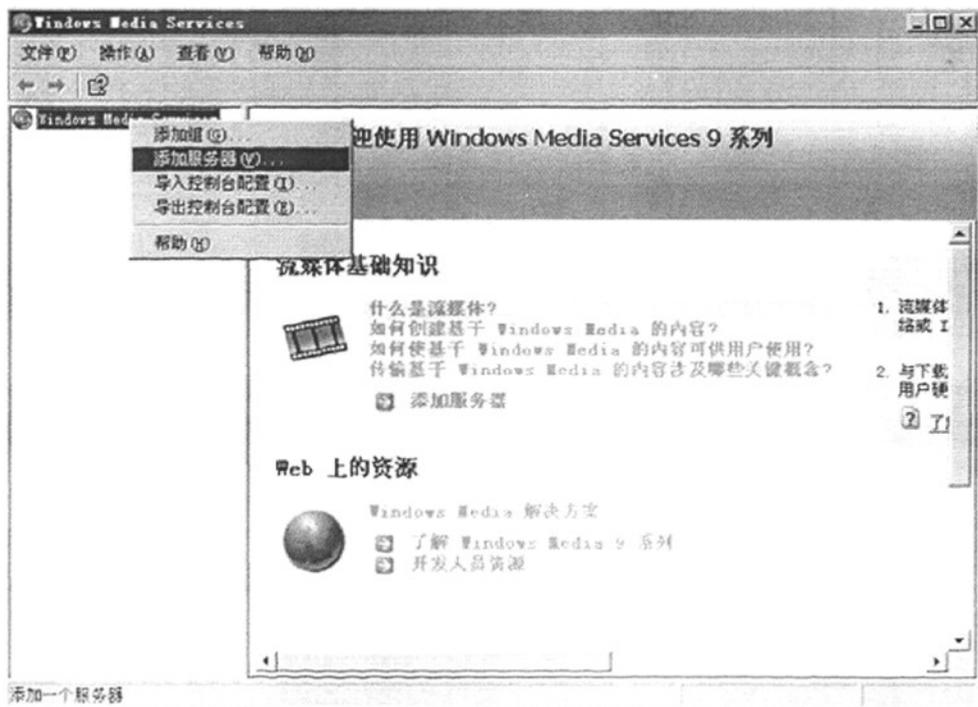
优点：直接可以支持 avi 格式的文件，无需漫长的文件转换过程。

缺点：技术不成熟，实现困难。

下面分别介绍两种方案的实现。

4.2.1 Windows Media Server 服务器的实现

这部分只需要简单的设置就可以实现，如下图 4-5:



4.2.2 简单视频点播服务器的实现

1. 服务器端界面实现

服务器端界面实现如图 4-6 所示:

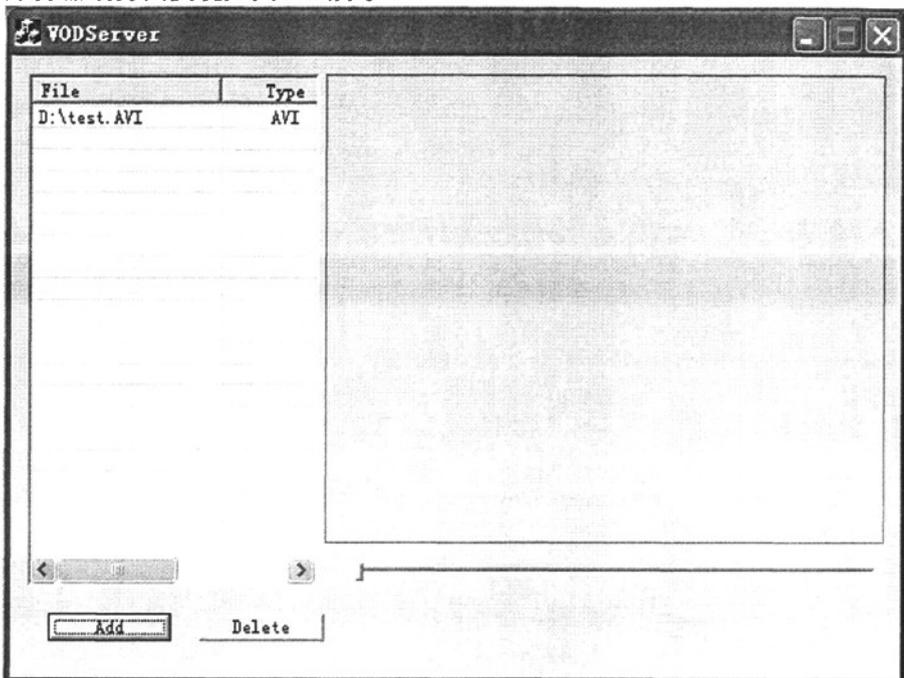


图 4-6 服务器端界面图

1) 服务器上提供给客户机的点播放节目列表

服务器收集和管理本地机器上的媒体文件，并维持一个节目列表供客户机点播。节目列表包含的信息有：媒体文件名，媒体文件类型，媒体文件的大小，该媒体节目的唯一数字代号（ProgramID,PID）。

2) 增加一个媒体文件到节目列表的按钮

单击本按钮将弹出一个标准的文件浏览对话框。用户可以浏览本地所有的媒体文件，并选择加入到节目列表中去。

3) 从节目列表中删除指定的媒体文件的按钮

用户首先在节目列表中选中想要删除的一项，然后单击本按钮即可。

2. 服务器端功能的实现

这部分分三个部分，控制部分，视频部分，音频部分。控制部分（对象 mControll）负责客户端的连接和命令的响应，视频部分（对象 mVideo）负责发送相应的视频流，音频部分（对象 mAudio）负责发送相应的音频流。

1) 服务器建立服务，等待连接

2) 接受到客户端的命令, 并作出相应的响应

3. 客户端界面的实现

客户端界面如图 4-7 所示:

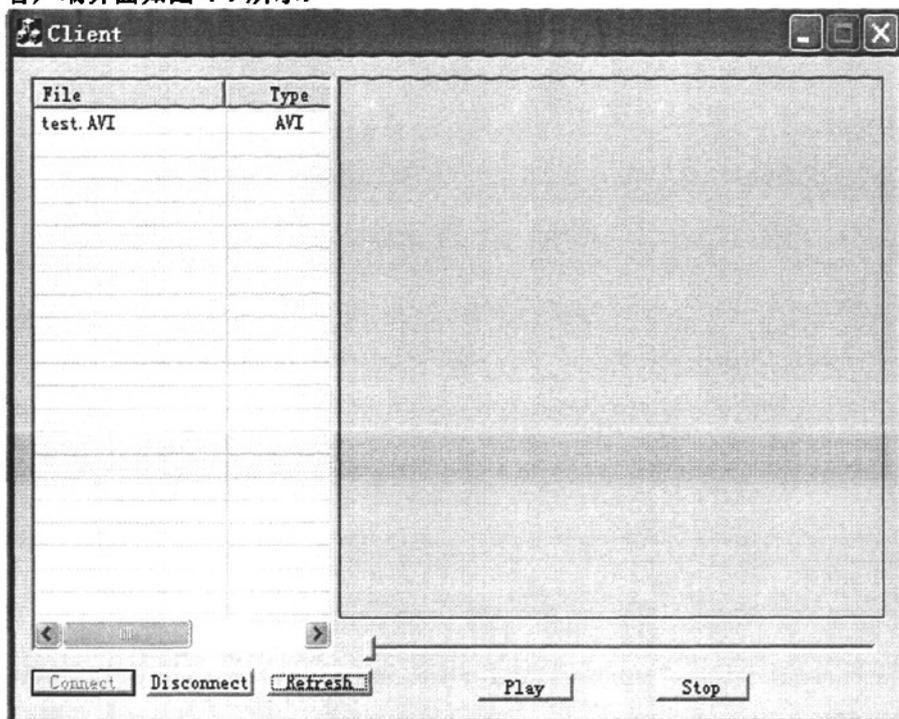


图 4-7 客户端界面图

1) 从服务器获得的可供点播的节目列表

当客户机与服务器建立网络通信连接之后, 就可以从服务器获取可供点播的节目列表。用户可以从该列表中随意悬着自己感兴趣的媒体节目进行播放。

2) 连接服务器

单击Connect按钮, 弹出一个对话框, 用户可以指定服务器的IP地址, 然后尝试与服务器建立网络通信连接。

3) 从服务器获取最新的节目列表

服务器提供点播的节目列表可能变动, 客户机提供Refresh按钮, 可以随时获取服务器上的最新的节目列表。

4) 指定的节目的播放

用户可以在客户机上显示的节目列表中随意选择一个自己感兴趣的媒体节目项, 然后单击Play按钮开始播放。

5) 停止节目的播放

4. 客户端功能的实现

这部分分三个部分，控制部分，视频部分，音频部分。控制部分（对象 mControll）负责客户端的连接和命令的响应，视频部分（对象 mVideo）负责发送相应的视频流，音频部分（对象 mAudio）负责发送相应的音频流。

- 1) 请求服务器连接，发送命令。
- 2) 接受服务器传送过来的音频视频流，并播放

4.3 本章小结

本章介绍了复合笔录和视频点播直播系统地实现。通过对上面复合笔录和视频直播点播的测试了解，我们可以知道本系统对于局域网用户来说，已经不成问题，当然还要有一个良好的服务网络。所以为了提供高质量和可靠的流媒体服务，在选择合适的视频直播点播服务器软件的同时还必须具备一个很好的服务网络。只有两者俱佳，我们才能搭建一个属于自己的流媒体服务平台

第五章 总结和展望

5.1 总结

该系统是本实验室在推进法庭数字化建设中的一个科研项目。旨在法庭相关人员停工一个面向应用的，逐渐能够满足相关人员不同的需要的，较为理想的数字化工作环境，同时能够为法官和书记员，庭审人员的交流提供一个互动平台。在此基础上，不断的深化改革，努力地促进庭审效率的提高。另外，通过这个品改，逐渐实现能够及时，准确地反馈庭审信息，提供一个较为科学的，日益完善的数字法庭系统，推动管理水平不断提高。

本人有幸参加了这个数字法庭系统地分析，设计与实现全过程，从最开始的用户需求，倒软件体系架构的选择，再到系统分析，系统设计，最后到系统实施和系统维护。这对自己而言是个非常难得的磨练机会，并且能够把多年积攒的理论知识应用于实践，哦诺这个项目，对软件工程的思想又新的，更高的体会，对于数字法庭系统的开发技术有了较大的提高，另外，也深感团队精神的重要性和必要性。

回顾整个研发过程，觉得这这也是一个难得的学习机会。本人在课题研究和探索过程中，查阅了大量的中英文的文献资料和相关课题的研究成果，充分的发挥了 Internet 这个取之不尽，用之不竭的资源宝库，访问了众多的相关网站，认真做了读书笔记和电子文档归类，积累了丰富的技术素材和技术储备，为最终完成毕业设计和毕业论文奠定了基础，真正实现了学以致用。

本系统的功能主要包括，庭审管理模块，复合笔录模块，视频点播直播模块等，并且基于角色实现系统功能。体现了数字法庭资源共享，交互式的特点。对未来数字法庭的发展作出了尝试性的研究。

目前本系统已经投入运行，效果良好，为法院的数字化建设作出了积极的贡献。需要特别指出的是，数字法庭系统是一个系统工程，也是一个不断完善的过程，尤其是一些数字化建设资源建设需要长期的积累，并且不断推陈出新。在此过程中，能够充分发挥计算机网络，多媒体技术，真正做到技术为应用服务，为人服务，以人为本。

当然由于时间仓促，人力，水平的局限，系统定会存在不少的缺点和不足，希望通过在实际的应用中不断完善，不断提高。

5.2 展望

1. 应用范围的扩展

本系统目前尚在局域网环境下运行，其应用规模和并发程度不是突出问题，随着信息化的应用范围的不断扩展。今后如果要扩展到外部网络中，面对数千人的访问量，可能需要采用更先进的技术才能解决问题。

2. 复合笔录部分的 OCX 封装技术的研究

本系统的复合笔录部分现在是以 .exe 文件的形式使用，下一步就是将其封装成 OCX 控件。

3. 视频点播直播部分需要做更深入的研究

现有的视频直播点播部分，在局域网内网速好的情况下，可以正常使用，将来要连接外部网络，网络如何优化，使视频流音频流正常播放需要研究。

参考文献

- [1] GAMMA E, HELM R, JOHNSON R, et al. Design patterns:Elements of reusable object-oriented software[M]. Addison Wesley, 1994.
- [2] 孙卫琴.精通 Struts:基于 MVC 的 Java Web 设计与开发[M]. 北京: 电子工业出版社,2004.
- [3] JOHNSON R, HOELLER J, ARENSEN A, et al. Java/J2EE application framework reference document. V1.1. 2004.
- [4] 徐长盛,戴超.一种快速开发 Web 应用程序方法的研究[J]. 计算机工程与设计,2004,(12):2237-2239.
- [5] 夏昕,曹晓钢,唐勇.深入浅出 Hibernate[M]. 北京: 电子工业出版社,2005.
- [6] JOHNSON R.Expert one-on-one J2EE design and development[M]. 魏海萍译.北京: 电子工业出版社,2003.
- [7] 任占勇, 周勇 Word 2000 中的 VBA 开发与应用[J],华北科技学院学报, 2003
- [8] 周宇, 黄春英, 龙侃 用 VBA 在 Word 中二次开发试题库.井冈山学院学报, 2005
- [9] WrightC.VisualC++程序员实用大全【M】.北京:中国水利水电出版社, 2001.
- [10] DavidJ.Kruglinski.VisualC 什技术内幕[M].北京:清华大学出版社, 1999.吴佑寿, 丁晓青.《汉字识别原理与应用》[M], 北京: 高等教育出版社, 1992.8.
- [11] Marsh K. Win32 Hooks[J/ CD] .MSDN ,2001 - 10
- [12] Walnum Clayton ,杜大鹏. Windows 2000 编程核心技术精解[M] . 北京:中国水利水电出版社,2002.
- [13] [美]Eugene Olafen ,等.MFC Visual C+ + 6 技术内幕[M] . 王建华等译. 北京:清华大学出版社,2000.
- [14] 戴春达, 符红光.Win32 中钩子的实现技术及其应用.计算机应用[J].2002,22(8).
- [15] Anthony Jones,Jim Ohlund,Windows 网络编程技术[M],北京机械工业出版社, 2001
- [16] Tom Armstrong,Ron Patton,ATL 开发指南[M],2 版,北京: 电子工业出版社, 2000
- [17] 马杰, 田金文. 流媒体技术及其文件格式[J].计算机工程与应用, 2003
- [18] Build the Filter Graph,<http://www.gdel.co.uk/building.htm>
- [19] Ze-Nian Li Mark S.Drew .多媒体技术教程 北京: 机械工业出版社, 2007
- [20] 陆其明.DirectShow 开发指南 北京: 清华大学出版社, 2003
- [21] 陆其明.DirectShow 实务精选 北京: 科学出版社, 2004
- [22] 郑键.基于 Web 的视频点播系统的设计与实现 中央广播电视大学出版社

- [23] 张江山, 鲁平, 等. 视频会议系统及其应用[M]. 北京: 北京邮电大学出版社, 2002.01
- [24] 鲁士文. 计算机网络协议和实现技术[M]. 北京: 清华大学出版社 2000.07
- [25] 张丽. 流媒体技术大全. 北京: 中国青年出版社, 2001
- [26] 李凌. Winsock 2 网络编程实用教程. 北京: 清华大学出版社, 2003
- [27] 王荣生. V O D 系统性能的综合研究. 计算机工程. 2005
- [28] 张辉. 基于流媒体的 MPEG-4 解码技术. 计算机应用技术. 2003

致谢

在这里我要特别感谢课题期间所有给予我支持和帮助的老师、同学以及家人和朋友。

首先衷心感谢导师邓中亮教授的悉心关怀和指导。邓老师一丝不苟的治学态度、孜孜不倦工作精神和谦逊的为人品质令我钦佩,将是我工作学习的学习榜样。在本人论文工作期间,一直得到邓老师的关心、帮助和支持。籍此机会,向邓老师表示由衷的感谢。

感谢黄建明老师,感谢他在学业和论文上给我的指导。感谢师兄邹德君等,感谢他们给我的支持与帮助。

还要感谢我的全体同班同学,在这两年半的时间里,他们给了我很多的生活和学习上帮助关心,也给我留下了纯真的友谊和很多快乐的回忆。

在这里,我还要感谢丁秉辉等朋友,有了他们的鼓励和支持,我能够勇往直前去努力求知和认真地踏实地做好工作。

攻读学位期间发表的学术论文

- 【1】 李鹏云, 邓中亮, 丁秉辉。基于 HOOK 函数复合笔录关键技术的研究与实现。中国科技论文在线, 2009 年 1 月, 论文署名北京邮电大学。