



南开大学学位论文使用授权书

根据《南开大学关于研究生学位论文收藏和利用管理办法》，我校的博士、硕士学位获得者均须向南开大学提交本人的学位论文纸质本及相应电子版。

本人完全了解南开大学有关研究生学位论文收藏和利用的管理规定。南开大学拥有在《著作权法》规定范围内的学位论文使用权，即：(1)学位获得者必须按规定提交学位论文(包括纸质印刷本及电子版)，学校可以采用影印、缩印或其他复制手段保存研究生学位论文，并编入《南开大学博硕士学位论文全文数据库》；(2)为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆等场所提供校内师生阅读，在校园网上提供论文目录检索、文摘以及论文全文浏览、下载等免费信息服务；(3)根据教育部有关规定，南开大学向教育部指定单位提交公开的学位论文；(4)学位论文作者授权学校向中国科技信息研究所和中国学术期刊(光盘)电子出版社提交规定范围的学位论文及其电子版并收入相应学位论文数据库，通过其相关网站对外进行信息服务。同时本人保留在其他媒体发表论文的权利。

非公开学位论文，保密期限内不向外提交和提供服务，解密后提交和服务同公开论文。
论文电子版提交至校图书馆网站：<http://202.113.20.161:8001/index.htm>。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人同意遵守上述规定。本授权书签署一式两份，由研究生院和图书馆留存。

作者暨授权人签字： 沈宇祥

2010年5月23日

南开大学研究生学位论文作者信息

论文题目	基于软件工程设计方法的板卡式网络视频监控系统研究				
姓名	沈宇祥	学号	2120070272	答辩日期	2010年5月23日
论文类别	博士 <input type="checkbox"/> 学历硕士 <input type="checkbox"/> 硕士专业学位 <input type="checkbox"/> 高校教师 <input type="checkbox"/> 同等学力硕士 <input type="checkbox"/>				
院/系/所	信息技术科学学院	专业	控制理论与控制工程		
联系电话	13752670917	Email	shenyuxiang@mail.nankai.edu.cn		
通信地址(邮编)：南开大学西区公寓 3-1-104					
备注：				是否批准为非公开论文	否

注：本授权书适用我校授予的所有博士、硕士的学位论文。由作者填写(一式两份)签字后交校图书馆，非公开学位论文须附《南开大学研究生申请非公开学位论文审批表》。

南开大学学位论文使用授权书

根据《南开大学关于研究生学位论文收藏和利用管理办法》，我校的博士、硕士学位获得者均须向南开大学提交本人的学位论文纸质本及相应电子版。

本人完全了解南开大学有关研究生学位论文收藏和利用的管理规定。南开大学拥有在《著作权法》规定范围内的学位论文使用权，即：(1)学位获得者必须按规定提交学位论文(包括纸质印刷本及电子版)，学校可以采用影印、缩印或其他复制手段保存研究生学位论文，并编入《南开大学博硕士学位论文全文数据库》；(2)为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆等场所提供校内师生阅读，在校园网上提供论文目录检索、文摘以及论文全文浏览、下载等免费信息服务；(3)根据教育部有关规定，南开大学向教育部指定单位提交公开的学位论文；(4)学位论文作者授权学校向中国科技信息研究所和中国学术期刊(光盘)电子出版社提交规定范围的学位论文及其电子版并收入相应学位论文数据库，通过其相关网站对外进行信息服务。同时本人保留在其他媒体发表论文的权利。

非公开学位论文，保密期限内不向外提交和提供服务，解密后提交和服务同公开论文。

论文电子版提交至校图书馆网站：<http://202.113.20.161:8001/index.htm>。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人同意遵守上述规定。本授权书签署一式两份，由研究生院和图书馆留存。

作者暨授权人签字： 沈宇祥

2010年5月26日

南开大学研究生学位论文作者信息

论文题目	基于软件工程设计方法的板卡式网络视频监控系统研究				
姓名	沈宇祥	学号	2120070272	答辩日期	2010年5月23日
论文类别	<input checked="" type="checkbox"/> 博士 <input type="checkbox"/> 学历硕士 <input type="checkbox"/> 硕士专业学位 <input type="checkbox"/> 高校教师 <input type="checkbox"/> 同等学力硕士 <input type="checkbox"/>				
院/系/所	信息技术科学学院	专业	控制理论与控制工程		
联系电话	13752670917	Email	shenyuxiang@mail.nankai.edu.cn		
通信地址(邮编)：南开大学西区公寓 3-1-104					
备注：				是否批准为非公开论文	否

注：本授权书适用我校授予的所有博士、硕士的学位论文。由作者填写(一式两份)签字后交校图书馆，非公开学位论文须附《南开大学研究生申请非公开学位论文审批表》。

南开大学学位论文原创性声明

本人郑重声明：所提交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名： 沈宇祥

2010年5月23日

非公开学位论文标注说明

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。未经批准的均为公开学位论文，公开学位论文本说明为空白。

论文题目							
申请密级	<input type="checkbox"/> 限制(≤2年)	<input type="checkbox"/> 秘密(≤10年)	<input type="checkbox"/> 机密(≤20年)				
保密期限	20	年	月	日至20	年	月	日
审批表编号		批准日期	20	年	月	日	

限制★2年（最长2年，可少于2年）

秘密★10年（最长5年，可少于5年）

机密★20年（最长10年，可少于10年）

摘要

软件工程是指导计算机软件开发和维护的一门工程学科。软件工程采用工程的概念、原理、技术和方法来开发和维护软件。它涉及到设计模式、程序设计语言、面向对象程序设计、软件测试等多方面知识。面向对象程序设计方法使软件开发与人类习惯的思维方式一致、开发的产品稳定性高且易于维护。软件测试技术则帮助软件开发人员对产品进行功能和性能测试，以确保开发的产品符合需求。

网络视频监控系统是安全防范系统的重要组成部分，视频监控系统因其直观、快捷、功能强、信息存储量大而应用于超市、银行等商业场所，发挥了巨大作用。近几年，随着计算机、网络以及图像处理、传输技术的飞速发展，视频监控技术也有长足的发展。

本文研究了软件工程方法及板卡式网络视频监控系统的设计和实现，主要内容如下：

(1) 讨论了软件危机的背景下形成的软件工程的涵义、基本原理、发展方向等概念。研究了设计模式的定义、分类等相关问题。本文还介绍了UML统一建模语言、软件测试及软件维护等方法。

(2) 本文结合当前网络视频监控系统的发展现状，以软件开发周期为主线，详细介绍了板卡式网络视频监控系统的设计和实现过程，我们依据“低耦合、高内聚”的设计思想采用了三层C/S体系结构设计系统，最大程度地保证了系统的可维护性和可扩展性。

(3) 本文结合监控现场软硬件环境，运用面向对象程序设计方法提高系统的易操作性、运用软件测试、软件维护和多种设计模式方法优化了系统性能，给出符合市场需求的解决方案，为相关软件设计理论的完善和在实际中的应用做出了有益的尝试。

关键词：软件工程，设计模式，面向对象软件设计，统一建模语言，软件测试，视频监控

Abstract

Computer software engineering is an engineering discipline that guides software development and maintenance. It adopts concepts, principles, techniques and methods to create and maintain software, which involves the design patterns, programming language, object-oriented design, software testing and other aspects of knowledge. Object-oriented design makes software development consistent with the natural way of human thinking. The products it produced are of high stability and easy to maintain. Software testing helps software developer with performance testing and functional test, which ensures that the products developed will meet the market's demand.

Because of its intuitive, fast, strong function and large storage capacity, video surveillance system plays a significant role in large supermarkets, banks and other commercial establishments. In recent years, video surveillance technology makes great stride in pace with the rapid development of technologies of computer network and image processing and transmission.

This paper studies software engineering, the design and implementation of networked video surveillance system of Board-type. The main contents are as follows:

(1) This thesis discusses the concepts such as the meaning of software engineering and its basic principles and development direction in the context of the software crisis. The definition and classification of design patterns and introduces the Unified Modeling Language (UML), software testing, software maintenance and other relevant issues are also discussed in it.

(2) The main line of this thesis is the software development cycle combining together with the development status of surveillance system. It details the design and implementation of networked video surveillance system of Board-type. We have designed the 3tier C/S architecture, which give the greatest degree of assurance of a software system scalability and maintainability based on the design ideology of "low coupling, high cohesion".

(3) Under the actual situations of the project, this thesis uses OOP, software testing, software maintenance, design patterns and many other methods to optimize the products' performance, giving solutions to meet the demand of market, with the

goal of improving object-oriented design theory and making instructive experiments in practical applications.

Keyword: Software Engineering, Design Pattern, Object-Oriented Design(OOD), Unified Modeling Language (UML), Video Surveillance, Software Testing

目 录

摘要.....	I
ABSTRACT.....	II
第一章 绪论.....	1
第一节 软件危机.....	1
第二节 视频监控系统的的发展历程及方向.....	2
第三节 本文的结构和内容安排.....	5
第二章 软件工程及面向对象软件设计开发方法.....	6
第一节 软件工程的相关理论基础.....	6
2.1.1 软件工程的基本概念.....	6
2.1.2 软件工程的方法论.....	7
2.1.3 软件工程系统的原则.....	7
2.1.4 软件工程的目标.....	9
第二节 软件生存周期.....	11
第三节 设计模式.....	11
2.3.1 设计模式的定义.....	11
2.3.2 设计模式的分类.....	13
第四节 UML 统一建模语言.....	14
第三章 系统的开发背景及可行性分析.....	16
第一节 视频监控系统.....	16
3.1.1 网络视频监控系统的组成.....	16
3.1.2 前期工作.....	18
第二节 需求分析及可行性研究.....	21
3.2.1 板卡式网络视频监控系统的优点.....	21
3.2.2 软件需求分析.....	22
第三节 开发模型.....	24
第四章 面向对象软件设计方法在板卡式视频监控系统中的应用 ...	26
第一节 软件的概要设计和详细设计.....	26

目录

4.1.1	软件总体设计.....	26
4.1.2	详细设计.....	28
第二节	具体功能的实现.....	30
4.2.1	预览.....	30
4.2.2	录像与回放.....	34
4.2.3	云台控制.....	37
4.2.4	移动侦测与遮挡.....	39
4.2.5	电子地图.....	40
第三节	技术难点.....	43
4.3.1	配置信息的存储.....	43
4.3.2	性能优化.....	46
4.3.3	移动侦测、遮挡功能中区域位置的存储.....	47
4.3.4	录像远程查询和下载.....	48
第四节	软件测试与软件维护.....	49
4.4.1	软件测试.....	49
4.4.2	软件维护.....	52
第五章	总结与展望.....	55
	总结.....	55
	展望.....	55
	参考文献.....	57
	致 谢.....	59
	个人简历、在学期间发表的学术论文与研究成果.....	60
	个人简历.....	60
	研究成果.....	60

第一章 绪论

第一节 软件危机

上世纪60年代中期，大容量、高速度计算机的出现，使计算机的应用范围迅速扩大，软件开发规模急剧增长。软件系统的规模越来越大，复杂程度越来越高，软件可靠性问题也越来越突出。原来的个人设计、个人使用的方式不再能满足要求，迫切需要改变软件生产方式，提高软件生产率，软件危机开始爆发。为了解决这一问题，诞生了软件工程这一学科。

所谓“软件危机”，是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。软件危机主要包含两方面的问题：一是如何开发软件以满足对软件日益增长的需求；二是如何维护数量不断增长的已有软件^[1]。

软件危机的主要表现形式如下：

(1) 软件开发进度难以预测。经费预算经常突破，完成时间一拖再拖，这种现象降低了软件开发组织的信誉；而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量，从而不可避免地会引起用户的不满。

(2) 用户对产品功能难以满足。开发人员和用户之间很难沟通、矛盾很难统一。往往是软件开发人员不能真正了解用户的需求，而用户又不了解计算机求解问题的模式和能力，双方无法用共同熟悉的语言进行交流和描述。在双方互不了解的情况下，就仓促上阵设计系统、匆忙着手编写程序，这样“闭门造车”的开发方式必然导致最终的产品不符合用户的实际需要。

(3) 软件产品质量无法保证。软件是逻辑产品，质量问题难以以统一的标准度量，因而造成质量控制困难。软件产品并不是没有错误，而是盲目检测很难发现错误，而隐藏下来的错误往往是造成重大事故的隐患。

(4) 软件产品难以维护。软件产品本质上是开发人员的代码化的逻辑思维活动，他人难以替代。除非是开发者本人，否则很难及时检测、排除系统故障。为使系统适应新的硬件环境，或根据用户的需要在原系统中增加一些新的功能，又有可能增加系统中的错误。

(5) 软件成本在计算机系统总成本中所占的比例逐年上升，软件的发展跟不上硬件的发展。由于微电子技术的进步和生成自动化程度不断地提高，硬件成

本逐年下降，然而软件开发需要大量人力，软件成本也随着通货膨胀以及软件规模和数量的不断扩大而持续上升。

(6) 软件开发生产率提高的速度远远跟不上计算机应用速度普及深入的趋势。软件的发展跟不上用户的要求，软件产品“供不应求”的现象使人类不能充分利用现代计算机硬件提供的巨大潜力。

造成上述软件危机的原因与软件自身的特点有关，也与软件开发人员在开发和维护时所采用的生产方式、方法和技术有关。这些原因可概括为以下几方面^[2]：

(1) 软件是计算机系统逻辑部件，软件产品往往规模庞大，结构复杂，这给软件的开发和维护带来客观的困难。

(2) 软件开发的管理困难。由于软件规模大、结构复杂，又具有无形性，导致管理困难、进度控制困难、质量控制困难，可靠性无法保证。

(3) 软件开发费用不断增加，维护费用急剧上升，直接威胁计算机应用的扩大。

(4) 软件开发技术落后。人们注重如编译原理、操作系统原理、数据库原理等一些计算机理论问题的研究，而不注重软件开发技术的研究。

(5) 生产方式落后。一方面，软件规模与复杂性增长了几个数量级，但生产方式仍然采用个体手工方式开发，根据个人习惯爱好工作，无章可循，无规范可依靠，带有很强的“个性化”特征的程序，因缺乏文档而根本不能维护，加剧了供需之间的矛盾。

(6) 开发工具落后，生产效率提高缓慢。软件开发工具原始，没有出现高效率的开发工具，因而软件生产效率低下。另外，软件开发人员忽视需求分析的重要性，轻视软件维护也是造成软件危机的原因。

第二节 视频监控系统的的发展历程及方向

当今社会，视频监控技术在社会生活中起着越来越重要的作用，它是安全保卫系统的重要组成部分，是一种功能强大的综合系统。网络视频监控系统以其准确、便捷、信息存储内容丰富而广泛应用于生产管理、安全防卫等多种场合。近年来，随着计算机、网络、图像处理和传输技术等的飞速发展，将监控网络并入计算机网络，以计算机网络为基础的网络视频监控系统可以在更大规

模更远距离内完成视频监控功能^[3]。

1.2.1 视频监控系统的的发展历程:

视频监控随着社会的进步,近来越来越被广泛地应用到各个领域。同时随着社会的发展,视频监控系统也随之经历了三个时代。

第一代:模拟时代。视频以模拟方式采用同轴电缆进行传输,并由控制主机进行模拟处理。

第二代:半数字时代。视频以模拟方式采用同轴电缆进行传输由多媒体控制主机或硬盘录像主机(DVR)进行数字处理与存贮。

第三代:全数字时代,视频从前端图像采集设备输出时即为数字信号,并以网络为传输媒介,基于国际通用的TCP/IP协议,采用流媒体技术实现视频在网上的多路复用传输,并通过设在网上的网络虚拟(数字)矩阵控制主机(IPM)来实现对整个监控系统的指挥、调度、存贮、授权控制等功能。此外报警、门禁、巡更等前端设备输出的数字信号也可由多网合一的方式通过网络复用进行传输并在同一平台上进行管理与控制。

现今视频监控系统已经步入了全数字时代。这将彻底打破“闭路电视系统”模拟方式的结构,从根本上改变了视频监控系统从信息采集、传输处理、系统控制的方式和结构形式,也标志着监控正在走向现代“四化”阶段即:

前端——一体化:监控系统前端一体化意味着多种技术的整合、嵌入式构架、适用和适应性更强以及不同探测设备的整合输出,为系统集成化奠定了基础。

传输——网络化:视频监控系统的网络化意味着系统的结构将由集总式向集散式系统发展,集散式系统采用多层分级的结构形式,将使整个网络系统硬件和软件资源以及任务和负载得以共享,这也是系统集成与整合的重要基础。

处理——数字化:信息处理数字化意味着信息流的数字化、编码压缩、开放式的协议,具有微内核技术的实时多任务、多用户、分布式操作系统,以实现抢先任务调度算法的快速响应,硬件和软件采用标准化、模块化和系列化的设计,系统设备的配置具有通用性强、开放性好、系统组态灵活、控制功能完善、数据处理方便、人机界面友好以及系统安装、调试和维修简单化,系统运行互为热备份,容错可靠等功能。

系统——集成化:系统集成化正是由于构建系统的各子系统均实现了网络

化和数字化，特别是使视频监控系统与弱电系统中其它各子系统间实现无缝连接，从而实现了在统一的操作平台上进行管理和控制。

1.2.2 视频监控系统的的发展方向：

前端一体化、视频数字化、监控网络化、系统集成化是视频监控系统公认的发展方向，而数字化是网络化的前提，网络化又是系统集成化的基础，所以，视频监控发展的最大两个特点就是数字化和网络化。

数字化：

数字化是21世纪的特征，是以信息技术为核心的电子技术发展的必然，数字化是迈向成长的通行证，随着时代的发展，我们的生存环境将变得越来越数字化。

视频监控系统的数字化首先应该是系统中信息流（包括视频、音频、控制等）从模拟状态转为数字状态，这将彻底打破经典闭路电视系统是以摄像机成像技术为中心的结构，根本上改变视频监控系统从信息采集、数据处理、传输、系统控制等的方式和结构形式。信息流的数字化、编码压缩、开放式的协议，使视频监控系统与安防系统中其它各子系统间实现无缝连接，并在统一的操作平台上实现管理和控制，这也是系统集成化的含义。

网络化：

视频监控系统的网络化将意味着系统的结构将由集总式向集散式系统过渡。集散式系统采用多层分级的结构形式，具有微内核技术的实时多任务、多用户、分布式操作系统以实现抢先任务调度算法的快速响应。组成集散式监控系统的硬件和软件采用标准化、模块化和系列化的设计，系统设备的配置具有通用性强、开放性好、系统组态灵活、控制功能完善、数据处理方便、人机界面友好以及系统安装、调试和维修简单化，系统运行互为热备份，容错可靠等优点。系统的网络化在某种程度上打破了布控区域和设备扩展的地域和数量界限。系统网络化将使整个网络系统硬件和软件资源的共享以及任务和负载的共享，这也是系统集成的一个重要概念。

综上所述，随着计算机技术及网络技术的迅猛发展，世界掀起一股强大的数字化、网络化浪潮，对于公安、安防行业的发展，必须经历模拟数字混合的阶段，但是最终的趋势必然是全面数字化，即视频在前端进行数字化、网络化，利用网络进行传输，采用分布式存储系统，模块化结构，完成视频监控任务。

为大型监控系统提供基于网络的解决方案，是目前市场的迫切需求。受天津某公司的委托，我们为该公司设计完成了一套板卡式网络视频监控系统的解决方案，并成功投入市场。

在这套系统设计之初，我们对市场做了调研，综合考虑了各种因素。为了增加软件的易维护性、可扩展性和易用性，我们采用快速原型法的开发模式；由于该解决方案主要在Windows系统上运行，采用了市面上比较流行的VS.NET 2003开发环境；为了兼容多家厂商的硬件设备，我们采用了多层架构，屏蔽了底层硬件的差异，对用户做到透明。

第三节 本文的结构和内容安排

本文研究了面向对象软件设计、软件工程等相关理论，并将软件复用、设计模式、软件测试等技术用到视频监控系统的开发和设计。

全文共分四章，结构安排如下：

第一章为绪论，首先介绍了软件危机的表现形式和产生的原因；然后介绍了近些年视频监控系统的发展状况。

第二章详细介绍了软件工程、软件生存周期、设计模式、UML统一建模语言等软件开发技术，为后续的说明工作做了理论铺垫。

第三章首先介绍了网络视频监控系统的组成结构，然后对系统进行需求分析及可行性研究，最后介绍了系统的开发模型。

第四章讲述了系统的具体实现过程，包括一些重点难点问题的解决思路，详细说明了面向对象程序设计、设计模式、体系结构设计、软件复用技术和软件测试技术的应用情况。

第五章总结了全文的工作，指出了本文的创新之处，最后给出了今后的工作方向。

第二章 软件工程及面向对象软件设计开发方法

软件工程是研究和应用如何以系统性的、规范化的、可量化的过程化方法去开发和维护软件，以及如何把经过时间考验而证明为正确的管理技术和当前能够得到的最好的技术方法结合起来^[4]。本章主要介绍了软件工程的相关理论。

第一节 软件工程的相关理论基础

2.1.1 软件工程的基本概念

“软件工程”一词是由北大西洋公约组织（North Atlantic Treaty Organization, NATO）的计算机科学家在联邦德国召开的国际会议上首次提出来的。产生软件工程这门学科的时代背景是“软件危机”。软件工程的发展和应用不仅缓和了软件危机，而且促使一门新兴的工程学科诞生了。

程序是为了解决某个特定问题而用程序设计语言描述的适合计算机处理的语句序列。软件不是程序，而是程序、数据以及开发、使用和维护程序需要的所有文档的完整集合。

软件工程是指导计算机软件开发和维护的一门工程学科。以下是两种比较典型的定义。

(1) 1986年，在第一届NATO会议上曾经给出了软件工程的一种早期定义：“软件工程就是为了经济地获得可靠的且能在实际机器上有效运行的软件，而建立和使用的完善的工程原理。”这个定义不仅指出了软件工程的目的是经济地开发出高质量的软件，而且强调了软件工程是一门工程学科，它应该建立并使用完善的工程原理。

(2) 1993年，IEEE进一步给出了一个更全面、更具体的定义：“软件是：a 把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；b 研究a中提到的途径。”

虽然软件工程的定义使用了不同的词句，强调的重点也有所差异，但是它的中心思想是把软件当作一种工业产品，要求“采用工业化的原理和方法对软件进行计划、开发和维护。”宗旨是为了提高软件生产率，降低生产成本，

以较小的代价获得高质量的软件产品。

2.1.2 软件工程的方法论

软件工程方法论中包含三个要素：方法、工具和过程。方法是完成软件开发的各项任务的技术手段，即解决“怎样做”的问题；工具指运用方法时使用的自动或半自动的支撑环境；过程是指为了获得高质量的软件所需要完成的一系列任务的框架，即规定完成各项任务的工作步骤^[5]。

软件工程方法论中目前使用最广泛的有两种方法：传统方法学和面向对象方法学。另外还有快速原型法、基于agent的方法、基于asp的方法等。

传统方法学也称为生命周期方法学或结构化范型。它是采用结构化技术（分析、设计和实现）来完成软件开发任务，并使用适当的软件工具和软件工程环境来支持该技术的运用。

传统方法学的特点是把软件生命周期的全过程依次划分为若干个阶段，然后顺序地完成每个阶段的任务。

面向对象的方法是和软件开发技术一起发展而来的一种方法，它从尽量模拟人类习惯的思维方式出发，使描述问题的空间（问题域）与实现解法的解域空间（求解域）在结构上尽可能一致。

面向对象方法学具有四个要点：把对象作为融合了数据及对数据操作行为的统一的软件构件；把所有对象都划分成类；按父类（基类）与子类（派生类）的关系，把若干相关的类组成层次结构的系统；对象间仅通过消息互相联系。

2.1.3 软件工程系统的原则

原则是言行所依据的准则，是人们行动的纲领。它告诉人们哪些事情可以做，哪些事情不能做^[6]。本节只做简单的讨论。

(1) 目的性原则：

目的是行动和努力要达到的目标，它决定着人们行动的方向。软件工程系统本身作为一个系统，它具有明确的目的：开发出满足客户需求的软件产品，同时要兼顾时间、成本和质量。

在软件工程系统中的一切活动也应以此为目标，偏离这一目标的活动将是错误的，是系统所不允许的。在软件工程系统中，这应当是我们的首要原则。

软件工程过程，就是要使用已有的成熟的解决问题的手段和方法来完成软

件产品的开发，不要试图将开发过程变为学习技术的过程。

(2) 实事求是原则：

实事求是是指从实际对象出发，探求事物的内部联系及其发展的规律性，认识事物的本质。通常指按照事物的实际情况办事。

在软件工程系统中同样要求我们实事求是地处理相关的问题，对事物进行更好的分析和理解，并应用于其各个环节的活动中。

例如，对于用户的需求，我们不能想当然地认为客户喜欢什么样的界面或功能，而是应当对客户有一个全面的评估和认识。在儿童教育多媒体软件中采用树形目录是不可取的。

同时在处理问题和解决问题的手段和方法上，也一定要实事求是，可以用简单方法解决的问题，就应当用简单的方法来处理，不应当对简单的事情复杂化。有时甚至是故弄玄虚，故意让人看不懂，这都是不可取的。

当然，实事求是的原则还包括我们对相关理论的研究上。

(3) 确定化原则

计算机所能够处理的数据只能是确定的数据，处理的事务也只能是确定的事务，任何不确定的输入对计算机而言都是不能接受的。软件是运行在计算机上的，它的所有部分都应当是确定的。人工智能或模糊化处理也只是用相应的确定的、形式化的算法来实现的。

计算机的这种确定性，要求我们在软件工程系统中也应当遵循确定化的原则。无论是问题的提出、分析，或是软件的构建、算法的实现，无一不要求我们对其确定化。

当然，在软件工程系统中有许多不确定的因素，对于这样的情况，也应当对其使用“确定化的语言”将这些“不确定的因素”描述出来。

(4) 适用性原则

适用性原则要求在软件工程系统中所做的选择符合目的性原则的前提下是适用的。有人喜欢用先进的技术手段和方法解决问题，这是不可取的。对于问题的解决和处理、开发工具的选择等，只要其适用就可以了。

在软件工程系统中对于开发工具的选择，不是先进的就是好的选择。只要能够实现我们的开发目标的开发都可以去采用，同时应当采用可能带来成本较低的开发工具。例如一个中等规模企业的进销存系统，不一定非要做成B/S模式，也不一定非要采用Java来进行开发。

2.1.4 软件工程的目標

软件工程的目標是：在给定成本、进度的前提下，开发出具有可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性并且满足用户需求的软件产品。追求这些目标有助于提高软件产品的质量和开发效率，减少维护的困难^[7]。

(1) 可修改性 (Modifiability)。

允许对系统进行修改而不增加原系统的复杂性。它支持软件的调试与维护，是一个难以达到的目标。

(2) 有效性 (Efficiency)。

软件系统能最有效地利用计算机的时间资源和空间资源。各种计算机软件都将系统的时/空开销作为衡量软件质量的一项重要技术指标。在很多场合，追求时间有效性和空间有效性方面会发生矛盾，这时不得不牺牲时间效率换取空间有效性或牺牲空间效率换取时间有效性，因此时/空折衷会经常出现。有经验的软件设计人员会巧妙地利用折衷概念，在具体的物理环境中实现用户的需求和自己的设计。

(3) 可靠性 (Reliability)。

能防止因概念、设计和结构等方面的不完善造成的软件系统失效，防止因操作不当造成软件系统失效。对于实时嵌入式计算机系统，可靠性是一个非常重要的目标，因为软件要实时地控制一个物理过程，如宇宙飞船的导航、核电站的运行等。如果可靠性得不到保证，一旦出现问题后果可能是灾难性的，不堪设想。因此在软件开发、编码和测试过程中，必须将可靠性放在重要地位。

(4) 可理解性 (Understandability)。

系统具有清晰的结构，能直接反映问题的需求。可理解性有助于控制软件系统的复杂性，并支持软件的维护、移植或重用。

(5) 可维护性 (Maintainability)。

软件产品交付用户使用后，能够对它进行修改，以便改正潜伏的错误，改进性能和其他属性，使软件产品适应环境的变化。由于软件是逻辑产品，只要用户需要，它可以无限期的使用下去，因此软件维护是不可避免的。软件维护费用在软件开发费用中占有很大的比重。可维护性是软件工程中一项十分重要的目标。软件的可理解性和可修改性有利于软件的可维护性。

(6) 可重用性 (Reusability)。

概念或功能相对独立的一个或一组相关模块定义为一个软部件。软部件可以在多种场合应用的程度称为部件的可重用性。可重用的软部件有的可以不加修改直接使用，有的需要修改后再用。可重用软部件应具有清晰的结构和注解，应具有正确的编码和较低的时/空开销。各种可重用软部件还可以按照某种规则存放在软部件库中，供软件工程师选用。可重用性有助于提高软件产品的质量和开发效率、有助于降低软件的开发和维护费用。从更广泛的意义上理解，软件工程的可重用性还应该包括：应用项目的重用，规格说明（也称为规约）的重用，设计的重用，概念和方法的重用，等等。一般来说，重用的层次越高，带来的效益也就越大。

(7) 可适应性 (Adaptability)。

软件在不同的系统约束条件下，使用户需求得到满足的难易程度。适应性强的软件应采用广为流行的程序设计语言编码，在广为流行的操作系统环境中运行，采用标准的术语和格式书写文档。适应性强的软件较容易推广使用。

(8) 可移植性 (Portability)。

软件从一个计算机系统或环境搬到另一个计算机系统或环境的难易程度。为了获得比较高的可移植性，在软件设计过程中通常采用通用的程序设计语言和运行环境支撑。对依赖于计算机系统的低级（物理）特征部分，如编译系统的目标代码生成，应相对独立、集中。这样，与处理机无关的部分就可以移植到其他系统上使用。可移植性支持软件的可重用性和可适应性。

(9) 可追踪性 (Tracability)。

根据软件需求对软件设计、程序进行正向追踪，或根据程序、软件设计对软件需求进行逆向追踪的能力。软件可追踪性依赖于软件开发各个阶段文档和程序的完整性、一致性和可理解性。降低系统的复杂性会提高软件的可追踪性。软件在测试或维护过程中或程序在执行期间出现问题时，应记录程序事件或有关模块中的全部或部分指令现场，以便分析、追踪产生问题的因果关系。

(10) 可互操作性 (Interoperability)。

多个软件元素相互通信并协同完成任务的能力。为了实现可互操作性，软件开发通常要遵循某种标准，支持折衷标准的环境将为软件元素之间的可互操作提供便利。可互操作性在分布计算环境下尤为重要。

第二节 软件生存周期

软件生存周期是借用工程中产品生存周期的概念而得来的，是指某一软件项目被提出来并着手实现开始直到该软件报废或停止使用为止。生存周期是软件工程的一个重要概念，把整个生存周期分为若干阶段，是实现软件生产工程化的重要步骤。

软件的生存周期一般分为软件计划、软件开发和软件运行三个时期。软件计划时期一般分为问题定义和可行性研究两个阶段；开发时期分需求分析、软件设计（包括概要设计、详细设计）、编码和测试四个阶段；运行时期主要是维护阶段。图2.1列出了一个典型的软件生存周期：

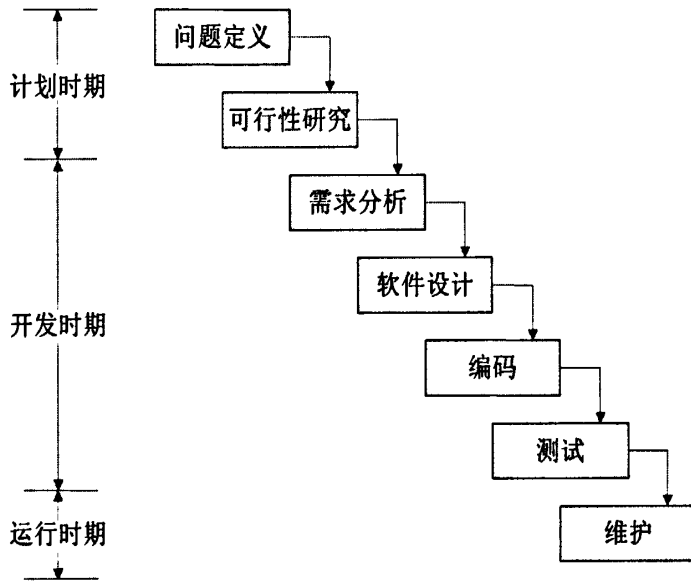


图2.1 软件生存周期示意图

第三节 设计模式

2.3.1 设计模式的定义

面向对象技术在近20年得到了日益广泛的应用，对经常重复出现的类似问题，运用面向对象技术已经总结出许多完善的解决方案，这些解决方案被称为设计模式^[8]。

模式的四个要素：

- (1) 模式名称：

它是一个助记名，使用一两个词描述模式问题、解决方案和效果。设计模式允许在较高抽象层次上进行设计。新模式的命名增添了设计词汇，在模式词汇表中，模式名有利于人们思考问题及相互交流，并可在书写文档时使用模式名。

(2) 问题：

它用于描述模式使用的场合和条件。描述设计的特定问题，解释设计问题存在的原因，描述导致不灵活设计的类或对象结构，有时还包含了模式应用必须满足的一系列先决条件。

(3) 解决方案：

它用于描述设计的组成部分、它们之间的相互关系及各自的职责和协作方式。由于模式像一个模板，可应用于多种不同场合，因此，解决方案并不描述一个特定而具体的设计或实现，而是提供设计问题的抽象描述和如何使用一个具体一般意义的元素组合来解决这个问题。

(4) 效果：

它描述了模式应用的效果及使用模式应权衡的问题。尽管我们描述设计决策时，并不总提到模式效果，但它们对于评价设计选择和理解使用模式的代价及好处具有重要意义。软件效果大多关注对时间和控件的衡量，它们也表述了语言和实现问题。由于软件重用是面向对象设计的要素之一，因此，模式效果包括它对系统的灵活性、扩充性或移植性的影响，显式地列出这些效果对理解和评价这些模式很有帮助。

不同的设计模式具有一些共性：

(1) 设计模式是从实践中获得的。

设计模式并非凭空想象产生的。实际上，它们大都来自于实践，是被“发现”而不是“写作”出来的。

(2) 设计模式表达好的设计。

设计模式并不仅是关于对象的设计，而且关于对象间的通信。因此有时也称为“通信模式”。正是简单而不乏优雅的通信方法设计使得设计模式变得如此重要。

(3) 设计模式避免无谓的重复劳动。

(4) 设计模式可重用。

(5) 多个设计模式可以一起用来解决更大的问题。

孤立地使用某个设计模式还不能够充分发挥设计模式的优势，通常将多个设计模式联合使用。

(6) 设计模式存在于不同层次的抽象级别。

设计模式存在于不同的粒度，小到具体的解决方案，大到通常的系统问题。

(7) 模式不断向前发展。

2.3.2 设计模式的分类

同可复用构件一样，设计模式在粒度和抽象层次上各不相同。由于存在众多的设计模式，我们希望用一种方式将它们进行分类。通过对设计模式进行分类，有利于使用者对各组相关的模式进行引用。分类有助于更快地学习模式，且对发现新的模式也有指导作用。在《设计模式—可复用面向对象软件的基础》书中，作者根据两条准则对模式进行分类^[9]。

(1) 第一是目的准则：

即模式是用来完成什么工作的。模式依据其目的可分为创建型(与对象的创建有关)、结构型(处理类或对象的组合)、或行为型(类或对象如何交互、如何分配职责)三种。

(2) 第二是范围准则：

指定模式主要是用于类还是用于对象。类模式处理类和子类之间的关系，这些关系通过继承建立，是静态的，在编译时刻便确定下来了。对象模式处理对象间的关系，这些关系在运行时刻是可以变化的，更具动态性。

对两条准则划分的设计模式的交叉部分说明如下：

创建型模式抽象了实例化过程。它们帮助一个系统独立于如何创建、组合和表示它们的那些对象。一个类创建型模式使用继承改变被实例化的类，而一个对象创建型模式将实例化委托给另一个对象。结构型类模式使用继承机制来组合类，而结构型对象模式则描述了对象的组装方式。行为型类模式使用继承描述算法和控制流，而行为型对象模式则描述一组对象怎样协作完成单个对象所无法完成的任务。

当然了，还有其它的分类型方式。从多角度思考模式有助于对它们的功能、差异和应用上下文有更深入的理解。

第四节 UML 统一建模语言

统一建模语言 (Unified Modeling Language, UML) 是一种编制软件蓝图的标准化语言, 它提供了描述软件系统模型的概念和图像的表示方法, 以及语言的扩展机制和对象约束语言。UML支持面向对象的技术和方法, 能够准确方便地表达面向对象的概念, 体现面向对象的分析和设计风格^[10]。

如本节前面的文字所述, 从软件的需求到软件的实现是一个十分复杂的过程。将软件的需求完全转化为计算机程序, 首先由用户提出软件需求的原始描述, 经过系统分析人员理解和分析, 形成软件需求文档。再经过软件开发人员的整体设计和详细设计, 并实现编程、测试, 最终得到符合用户需求的软件。从用户需求到程序的实现需要经过多个步骤, 其中包括大量人工的、创造性的、复杂的脑力劳动。

统一建模语言(UML)定义良好、易于表达、功能强大, 不仅支持面向对象的分析与设计, 而且支持从需求分析开始的软件开发的全过程。但如何恰当地将这种可视化图形建模技术用于解决软件开发所面临的问题, 如何研究和开发支持UML的的建模工具, 仍是目前该领域的热点问题。目前, 在基于UML的开发方法和环境方面, 国际上已经进行了一些研究和实际开发工作, 已形成产品的支持UML的工具具有IBM的RationalRose、Sybase的PowerDesigner、Gentleware的Poseidon、Visual Object Modelers的Visual UML等几十种优秀的工具, 它们都具有UML建模的基本功能以及众多强大的扩展功能^[11]。

RationalRose目前在面向对象分析、建模、设计工具市场上起着主导作用。其特点如下:

(1) 支持UML对象建模符号标准。RationalRose提供完整的UML支持。从系统分析、设计至程序设计, 均可用UML来表示。UML让系统开发的所有参与者, 在此世界通用的建模语言的基础上沟通。

(2) 支持构件化软件发展。对复杂系统而言, 构件式的软件开发已跃升为最有效率的模式。RationalRose允许使用者利用构件图(ComPoneniDiagram), 清楚地表达构件及其界面之间的关联。

(3) 支持多种语言的软件开发。在大型软件开发中, 使用数种的程序语言很常见的。RationalRose企业版提供了多种语言的支持, 让使用者可以发展不同程序语言的构件。RationalRose可从模型产生C++、Java、VB、Delphi、smallTalk或Ada的程序代码。

(4) 支持正向、逆向工程。RationalRose具有正向工程(ForwardEngineering)、反向程(ReverseEngineering)等功能特性,可以在修改实际程序后将最新的程序代码状况,迅速地反应到设计模型中,以保持对象设计模型与程序代码的一致性。

(5) 强力支持团队开发。无论团队有10或100个成员,RationalRose均能有效地协调他们之间的工作。

(6) 扩充性。RationalRose提供了较好的扩充性,使用者可透过RationalRose的OLEAutomationServe界面API,取得在RationalRose中系统开发者所设计的各种模型的资料,并加以运用,透过RationalRose外挂功能(Add-In),达到扩充RatinnalRose功能的目的。目前已有超过100家厂商通过此方式,提供各样的外挂构件,扩充Rational-Rose的功能,提供更多元化的服务。

第三章 系统的开发背景及可行性分析

上一章主要介绍了面向对象软件开发设计的一些基本理论。本章将针对板卡式网络视频监控系统进行需求分析和可行性分析。首先介绍一下系统的开发背景。

第一节 视频监控系统

3.1.1 网络视频监控系统的组成

第一章介绍了视频监控系统的历史和将来的发展方向，大家对监控系统有了一个整体的概念，本章将详细介绍监控系统的各个组成部分。

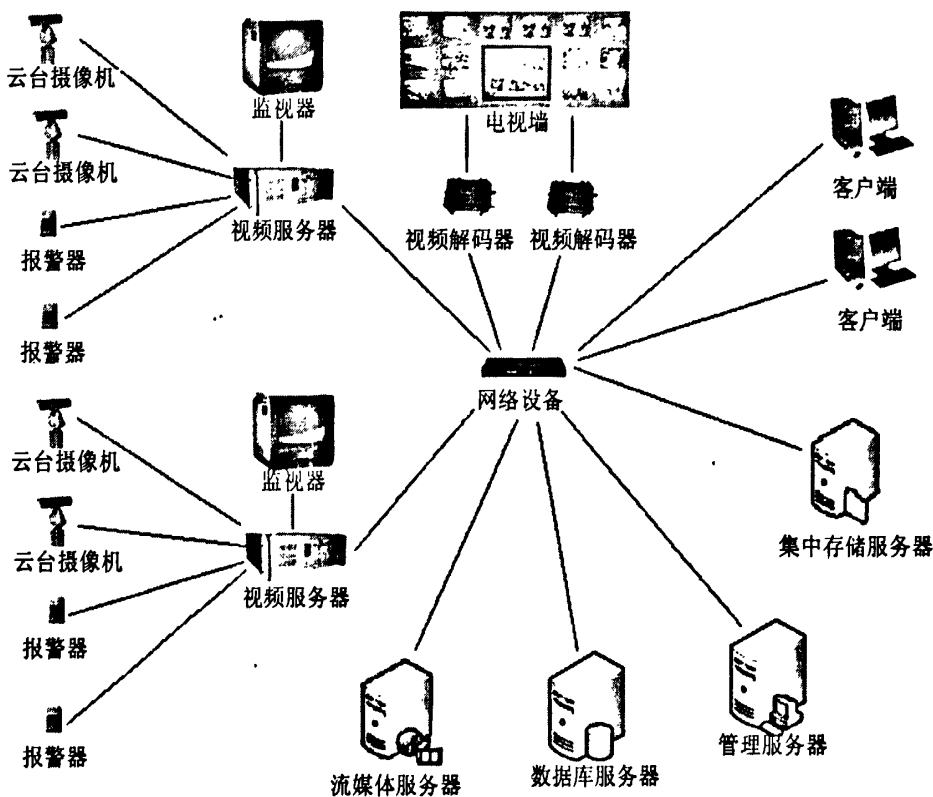


图3.1 网络视频监控系统的物理结构图

上图3.1是一个典型的网络视频监控系统的物理结构图。图中可以看到，云

台摄像机将采集到的各个通道的图像以模拟信号的形式发送给视频服务器（即硬盘录像机）；视频服务器将模拟信号压缩、编码，转化成数字信号，一方面存储在自身携带的大容量硬盘上，一方面发送到网络上；集中存储服务器也具备大容量硬盘，可以将这些数字信息记录下来，用于后期的录像查询；普通客户端接收到这些实时的数字信号，采用软件解码技术就可以浏览摄像机采集到的图像（如果客户端数量很多，可以增加流媒体服务器和流转发服务器来增加负载，从而方便更多用户的浏览）；如果用户想更清晰、更方便快捷地浏览通道信息，可以将网络上的数字信号经过视频解码器解码，重新转化为模拟信号投射到大屏幕的电视墙上。

网络视频监控系统中的主要组成部分及功能描述如下：

(1) 云台镜头控制系统：

云台镜头控制系统主要是由云台和控制器组成，用于完成从监控中心遥控摄像机观测位置的变动及观测点图像的放大、缩小处理。

(2) 信号传输系统：

在本系统中信号传输系统包括电源的传输、视频信号的传输和控制信号的传输三部分。

(3) 视频处理系统：

视频处理系统主要完成对视频信号的数字化处理（模拟信号转化成数字信号）、图像的显示、存储、及图像信号的远程传输。本系统中采用硬盘录像及远程传输系统实现对所有采集点通道的显示、录像的回放以及远程视频图像预览。

视频处理系统中处于核心部分的设备是数字视频录像机，相对于传统的模拟视频录像机，它是采用硬盘录像的，故又被称做硬盘录像机，也被称为（DigitalVideoRecorder）DVR。它是一套进行图像存储处理的计算机系统，具有对图像/语音进行长时间录像、录音、远程监视和控制的功能，DVR集合了云台镜头控制、网络传输、画面分割器、录像机、报警控制等五种功能于一身，一台设备就具有模拟监控系统一大堆设备的功能，并且在价格也很低廉。DVR采用的是数字记录技术，在图像处理、图像备份、储存、检索以及网络传递、远程控制等方面也远远优于模拟监控设备。DVR代表了电视监控系统的发展方向，已经成为目前市面上电视监控系统的首选产品。目前市面上流行的产品有PC式DVR和嵌入式DVR两种^[15]。

图3.2是嵌入式网络硬盘录像机实物图。

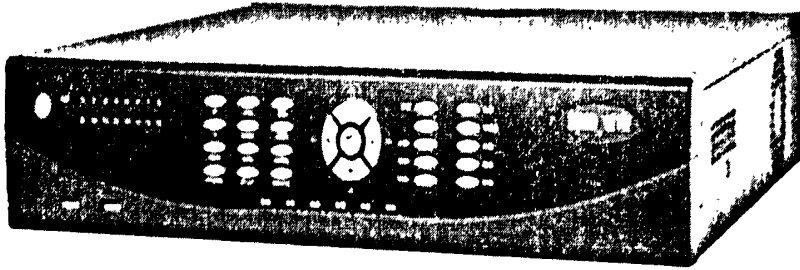


图3.2 嵌入式网络硬盘录像机（嵌入式DVR）

嵌入式DVR内含视频处理芯片、大容量硬盘等高性能设备，具有云台镜头控制、网络传输、画面分割、录像、报警控制等多种功能，由于其较高的稳定性和强大的功能，它在监控系统领域占据大部分市场份额。

3.1.2 前期工作

在本系统设计实现以前已经完成了嵌入式网络视频监控系统的设计开发工作，为了满足市场需求，要设计板卡式网络视频监控系统。

嵌入式系统指的是以应用为中心和以计算机技术为基础的，并且软硬件是可裁剪的，能满足应用系统对功能、可靠性、成本、体积、功耗等指标严格要求的专用计算机系统。简单地说，嵌入式系统集成系统的应用软件与硬件于一体，具有软件代码小、高度自动化、响应速度快等特点，特别适合于那些要求实时和多任务的体系结构，可以实现对其他设备的控制、监视或管理等功能。嵌入式系统因其体积小、可靠性高、功能强、灵活方便等许多优点，其应用已经深入到制造业。过程控制、网络、通信、仪器、汽车、船舶、航空、航天、军事装备、消费类产品等应用领域，对各行各业的技术改造、产品更新换代、加速自动化进程、提高生产率等方面起到了极其重要的推动作用^[16]。

可以从以下几方面来理解嵌入式系统：

(1) 嵌入式系统是面向用户、面向产品。面向应用的，它必须与具体应用相结合才会具有生命力，才更具有优势。嵌入式系统与应用紧密结合，具有很强的专用性，必须结合实际系统需求进行合理的裁剪、利用。

(2) 嵌入式系统是将先进的计算机技术、半导体技术、电子技术以及各个行

业的具体应用相结合后的产物，因此它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。

(3) 嵌入式系统必须根据应用需求对软硬件进行裁剪，满足应用系统的功能、可靠性、成本、体积等要求。如果能建立相对通用的软硬件基础，然后在其上开发出适应各种需要的系统，将会是一种比较好的发展模式。目前嵌入式系统的核心往往是一个只有几个KB到几十KB的微内核，需要根据实际应用进行功能扩展或者剪裁，而由于微内核的存在，使得这种扩展或剪裁能够非常顺利地进行。

嵌入式系统把系统的应用软件与硬件融于一体，有点类似于个人电脑中BIOS的工作方式，具有响应速度快、软件代码小、高度自动化等特点，特别适合于对实时和多任务要求较高的应用。嵌入式DVR就是基于嵌入式处理器和嵌入式实时操作系统的嵌入式系统，它采用专用数字编/解码芯片对图像进行压缩和解压，嵌入式操作系统主要是完成嵌入式整机的控制及管理。这类产品在设计制造时对软、硬件的稳定性进行了专门有针对性的规划，整个视音频的压缩、显示、网络等多种功能全部可通过一块单板来实现，大大地提高了整个系统的可靠性和稳定性，故障率很低。

在先前的工作中，已经设计实现了一套嵌入式网络视频监控系统，主要实现了以下两部分功能：

服务器(Server)：主要用于数据库配置，用户通过它添加各种资源，设置用户权限。主要功能：登录验证、操作日志、系统日志、报警日志的记录与查询、资源和用户的配置管理、预案管理，预览组管理，电视墙方案的管理[20]。

图3.3是服务器端软件界面图。

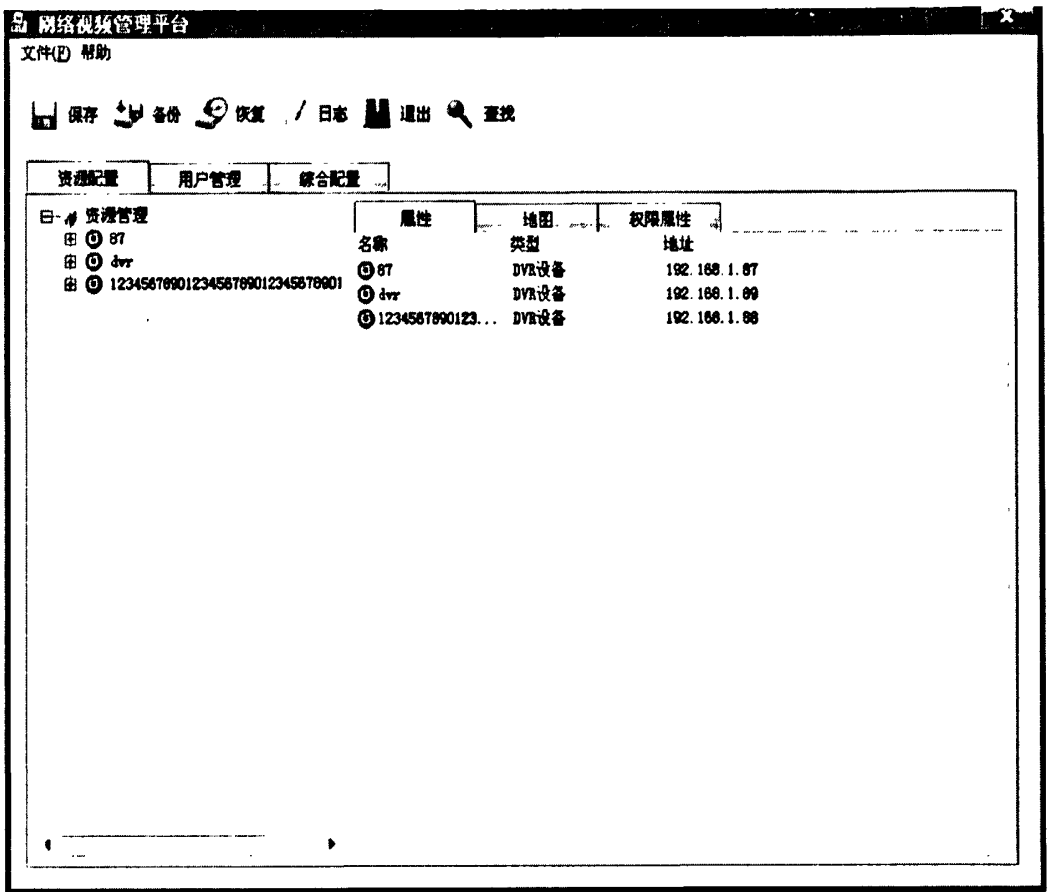


图3.3 服务器端界面图

客户端 (Client): Windows应用程序, 用户根据服务器端配置的用户名, 密码登陆后, 可以对所拥有权限的资源进行对应权限的操作。主要功能: 登录验证、操作日志、系统日志、报警日志的记录与查询、多路视频浏览、云台控制、录像、回放、电子地图, 报警、调用预览组、操作电视墙、设置多种信息。

图3.4是客户端软件界面图。



图3.4 客户端界面图

第二节 需求分析及可行性研究

3.2.1 板卡式网络视频监控系统的优点

上一节介绍了嵌入式视频监控系统，那为什么还要开发板卡式监控系统呢？与嵌入式相比它又有哪些优缺点呢？这里为大家详细介绍一下。

PC式硬盘录像机（PC式DVR）以传统的PC机为基本平台，以目前市面上主流操作系统为基本软件，装备图像采集压缩卡，编制软件成为一套完整的系统。PC机是一种非常通用的平台，PC机的硬件更新换代速度快，因此PC式DVR的产品性能提升很容易，于此同时软件修正、升级也十分方便。PCDVR各种视频监控功能的实现都依靠板卡来完成，比如视音频压缩卡、PC机自带的网卡、声卡、显卡等，这种插卡式的系统在系统装配、维修、运输中是非常容易出现不可靠的问题的，因此PC式DVR不能用于工业控制监控领域，只适合于对可靠性要求不高的但对图像质量要求较高的商用办公环境。

PC式DVR相对于嵌入式DVR的优点：

- (1) 存储空间很大（且可扩展），能够长时间录像，本地存储也很稳定。
- (2) 良好的人机接口和方便的文件管理，通过鼠标、键盘可以很直观地操作，

只要用过计算机的人都很容易上手。

(3) 软、硬件升级比较容易，产品更新快。

(4) 维修成本比较低，一般的故障都可以通过更换问题部件进行维修，整机不会报废。

(5) 图像清晰度高，即图像分辨率高，适用于银行、超市等对监控系统标准要求高的场所使用。

图3.5是视频压缩卡的示意图。

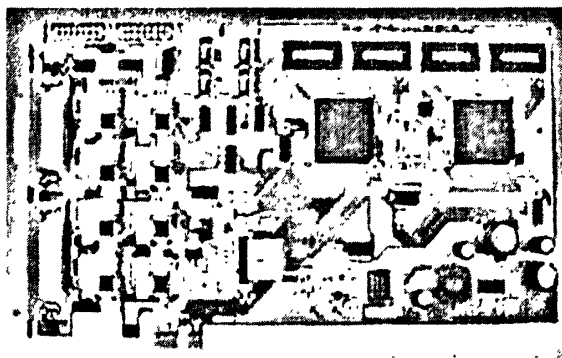


图3.5 PCI-E接口视音频压缩板卡

3.2.2 软件需求分析

软件需求分析是软件开发周期的第一个阶段，是整个系统开发的基础，是软件生存周期最重要的一步。需求分析的任务还不是确定系统怎样完成它的工作，而仅仅是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰而且具体的需求^[21]。

分析员和用户双方确定对软件系统有下述几方面的综合要求。

(1) 功能需求：功能需求指所开发软件系统必须提供的服务，划分出系统必须完成的所有功能，这是最重要的。

(2) 性能需求：性能需求指所开发的软件的技术性指标。通常包括存储容量、运行时间等限制。

(3) 环境需求：环境需求指软件运行时所需要的软、硬件（如机型、外设、操作系统和数据库管理系统）的要求。

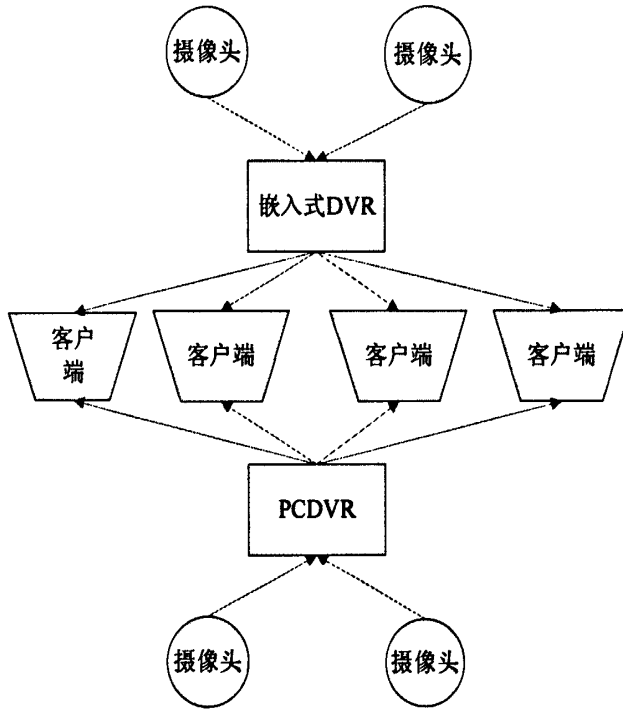
(4) 接口需求：接口需求描述应用系统与它的环境通信的格式。常见的接口需求有：用户接口需求、软件接口需求、意见接口需求和通信接口需求。

(5) 用户界面需求：用户界面需求即指人机交互方式、输入/输出数据格式

等。

另外，还有可靠性、安全性、保密性、约束、可移植性和可维护性等方面的需求，这些需求通常可以通过双方交流、调查研究来获取，并达成共识。

将PC式DVR加入到现有系统中的简略结构如图3.6所示。



3.6 系统结构图

结合实际系统，可以总结出具体的系统需求：

(1) 功能需求：网络视频监控系统所具备的功能有如下几方面，图像实时预览、录像、录像回放、录像下载、云台控制、移动侦测、遮挡、报警、电子地图。

(2) 性能需求：监控系统里涉及到的技术性指标是比较多的，例如：图像的分辨率、图像编码流类型、码流比特率、移动侦测的灵敏度等等。此外，系统要求全天24小时运行，还要保证程序不出故障，各项功能正常。

(3) 系统主要在Windows上运行，数据库采用的是免费的小型数据库MySQL，硬件设备是普通的PC机。

(4) 接口需求主要有PC机的COM口（串行通信接口）、LAN口等。

(5) 系统对用户界面的要求非常高，因为用户方操作这套系统的人员都不是专业的计算机技术人员，他们对监控系统里设备的参数、复杂的计算机操作方

式都不是很懂，因此系统操作必须简洁、人性化、甚至做到傻瓜式操作，对用户要屏蔽尽可能多的技术细节，让那些非专业人员也能很容易地操作、很直白地看到反馈的结果。

第三节 开发模型

建模是软件工程最常使用的一项技术。所谓软件开发模型，就是为整个软件生存期建立的模型。过去几十年，已先后出现了多种软件开发模型，包括传统模型、演化模型和面向对象模型等。它们虽各有特色，分别适用于不同特征的软件项目，但一般都包含“计划”、“开发”和“维护”三类活动。

“What-How-Change”概括了三类活动的主要特征，即在计划时期要弄清软件“做什么”；开发时期集中解决让软件“怎么做”；维护时期主要是对软件的“修改”。目前具体的软件开发模型包括几十种，如瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、构件集成模型、转换模型和净室模型等^[22]。

考虑到本系统的实际情况选用快速原型模型作为开发模型。下面简单介绍一下快速原型模型。

快速原型模型的主要思想是：首先快速建立一个能够反映用户主要需求的原型系统，让用户在计算机上试用它，通过实践让用户了解未来目标系统的概貌，以便判断哪些功能是符合需要的，哪些方面需要改进。用户会提出许多改进意见，开发人员按照用户的意见快速地修改原型系统，然后再次请用户试用……，这样反复改进，最终建立完全符合用户需求的新系统。图 3.7 是快速原型模型法的生存期模型。

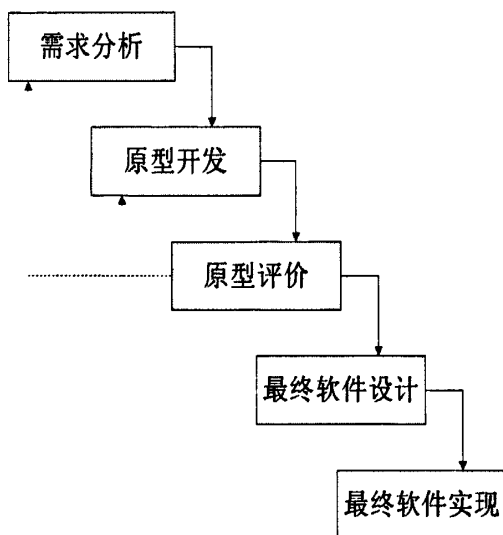


图3.7 快速原型模型法生存期模型

在软件开发过程中，要不要快速建立软件原型，这要视软件系统的性质和规模而定。当系统要求复杂，系统服务不太清楚时，在需求分析阶段开发一个软件原型是很值得的，可以大大减少因系统需求的可能性错误而导致的损失。特别是当性能要求比较高时，在软件原型上先做一些实验也是很必要的。

选择快速原型模型法的理由是其具有如下两个特点：

(1) 快速原型应具备的第一个特点是“快速”。快速原型的目的是尽快向用户提供一个可以在计算机上运行的目标系统的原型。

(2) 快速原型应具备的第二个特点是“容易修改”。系统原型建立后，让用户对原型进行试用评估，并提出意见，开发者根据用户的意见迅速修改原型，并构建原型的第二版，再让用户试用评估，开发者再根据用户意见修改。这样，“试用—评估—修改”过程可能重复多遍，直到用户和开发者都满意为止。

因此，快速原型开发模型符合系统设计的要求，用这种开发模型来开发这套系统是可行的。

第四章 面向对象软件设计方法在板卡式视频监控系统中的应用

这一章将继续软件开发周期的下一个阶段——详细介绍板卡式网络视频监控系统的的设计实现过程，说明设计模式、系统体系结构设计、软件测试和维护等技术在实际中的应用。

第一节 软件的概要设计和详细设计

经过需求分析阶段的工作，建立了系统的逻辑模型，系统必须“做什么”已经清楚了，下一步将进入软件设计阶段，即着手实现系统需求，要把“做什么”的逻辑模型变换为“怎样做”的物理模型。软件设计分两部分：总体设计和详细设计。总体设计是进入软件设计的第一个阶段，只描述软件的总的体系结构，第二个极端是详细设计，即对结构进一步细化。

4.1.1 软件总体设计

软件总体设计的主要任务就是软件结构的设计，为了提高软件设计质量，必须根据软件设计的原理改进软件设计。长期以来，人们在计算机软件开发的实践中积累了丰富的经验，总结这些经验可以得出以下6个软件结构设计准则^[24]。

(1) 低耦合，高内聚。

初步设计出软件结构后，为了提高模块的独立性，应该审查并分析软件结构，通过分解和合并模块降低耦合程度，提高内聚程度。

(2) 模块结构的深度、宽度、扇出和扇入应适当。

深度指软件结构中模块的层次数，它表示控制的层数，在一定意义上能粗略地反映系统的规模和复杂程度。

宽度指同一层次中最大的模块个数。它表示控制的总分布。

扇出是一个模块直接调用的模块数目。经验证明，好的系统结构的平均扇出数是3~4，最大不能超过9。

扇入是指有多少个上级模块直接调用它。一个模块的扇入越大，说明共享该模块的上级模块数目越大，这是有好处的。

一般设计得比较好的软件结构，顶层扇出高，中层扇出较少，底层模块高扇入。

- (3) 模块的作用范围应该在控制范围内。
- (4) 模块接口设计要简单，以便降低复杂程度和冗余度。
- (5) 设计功能可预测并能得到验证的模块。
- (6) 适当划分模块规模，以保持其独立性。

分层设计是软件设计中非常常见的设计方法。对于复杂问题，分解是行之有效的办法。这里采用三层客户机/服务器的体系结构。如图4.1所示。

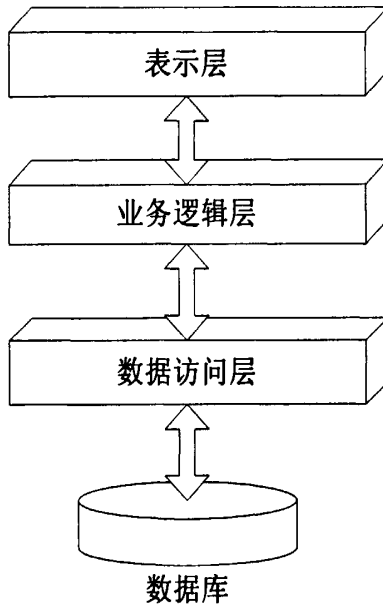


图4.1 三层客户机/服务器体系结构图

表示层是系统的用户接口部分，担负着用户与系统间的对话功能。表示层主要用于检查用户从键盘等设备输入的数据，显示底层返回的结果，检查的内容也只限于数据的形式和取值的范围，不包括有关应用本身的处理逻辑。它一般只与中间的逻辑层交互，而不直接访问后台数据库。

业务逻辑层又可以称为功能层，是应用处理的核心，是连接上层界面与底层的中介和桥梁。它相应用户发来的请求，执行相应的逻辑处理，并将底层的执行结果返回给客户端。这层中包含有确认用户对应用和数据库存取权限的功能，同时努力使表示层和业务逻辑层之间的数据交换尽可能地简单，做到信息一次性传输。

数据访问层的主要组成部分就是数据库管理系统，负责管理对数据库中数据的读写工作，该层的主要任务是实现数据的存储、数据的访问控制、数据完整性约束和并发控制等。

三层C/S体系结构具有以下优点：各个层次在逻辑上保持相对独立，系统的结构十分清晰，便于系统维护和扩展；允许各个层次的应用开发同时进行，缩短了整体软件的开发周期；功能层能够有效隔离界面层与数据访问层，从逻辑层面上增强了数据库的安全性。

4.1.2 详细设计

详细设计是软件设计的第二阶段，在此之前的总体设计阶段已将系统划分为多个模块，并将它们按照一定的原则组装起来，同时确定了每个模块的功能及模块与模块之间的外部接口。这一阶段的工作就是要对系统中的每个模块给出足够详细的过程性描述，故也称之为“过程设计”。

详细设计的根本目的就是确定应该怎样具体实现所要求的系统，也就是说，经过这一阶段的工作，应该得出对目标系统的精确描述，具体的就是为软件结构图中每一个模块确定采用的算法和块内数据结构，用某种选定的详细设计工具更清晰地描述，从而在编码阶段可以把这些描述直接翻译成某种程序设计语言书写的源程序。

详细设计阶段的任务是要设计出程序的“蓝图”，以后程序员将根据这个蓝图写出实际的代码。因此，详细设计的结果基本上决定了最终程序代码的质量。考虑程序代码质量时必须注意，程序的“读者”有两个，即计算机和人。在整个软件生存周期中，软件测试、诊断程序错误、修改和软件维护等都必须先读懂程序。实际上对于长期使用的软件系统，读程序的时间可能比写程序的时间要长得多。因此，衡量程序的质量不仅仅看它的逻辑是否正确，性能是否满足要求，更重要的是看它是否易读、易理解。详细设计的目的不仅仅是逻辑上正确地实现每个模块的功能，更重要的是设计的处理过程应该尽可能地简明易懂。结构程序设计技术是实现上述目的的关键技术，因此，是详细设计的逻辑基础。

因为程序主要使用厂家的SDK进行主要功能的实现，而不同厂家的不同设备的SDK有很大差异性，因此希望系统能够提供统一的接口来实现所需的功能。设计模式里的Facade模式可以解决这个问题。

Facade 模式的作用是为拥有相同功能的子系统提供一致的接口。Facade 模

式的结构图如图 4.2 所示。这个功能可以完全满足要求。因此这里采用 Facade 模式解决 SDK 封装的难题。这里用厂家提供的 SDK 实现了对 DVR 的基本操作，将这些基本操作封装到一个类中去，实现了一个高层接口，从而对客户屏蔽了复杂的 SDK^[26]。

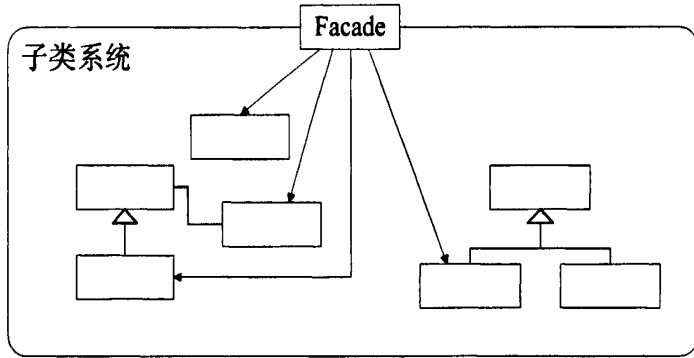


图4.2 Facade模式结构图

图4.3是设备类的派生关系图，大家可以看到，程序抽象出了所有DVR的共同特征（播放、录像、截图、云台控制、获取地址、获取设备类型、语音对讲等功能），然后封装到一个虚基类中，完全屏蔽了底层差异，使编程更加简单、方便。

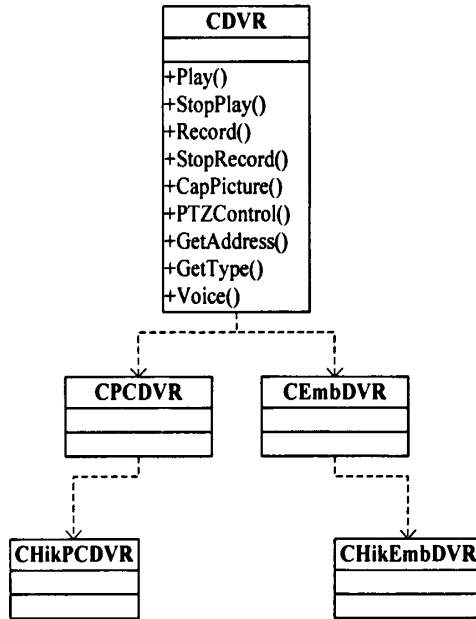


图4.3 设备类的继承派生示意图

其它具体的设计方法会在后续章节中详细介绍。

第二节 视频监控系统的功能和实现

4.2.1 预览

下面讲述一下播放控制逻辑的编写。如图4.4所示，界面右侧树形列表显示了整个监控系统的资源，用户可以点击每个资源前面的方框来打开或关闭该资源。用户也可以根据需要切换显示通道录像状态、报警状态、图像视频参数设定和云台设置界面。下侧是功能按钮，包括画面个数控制按钮、录像、回放、抓图、锁定、对讲、电子地图和系统设置。中间占据最大区域的是画面显示区域。

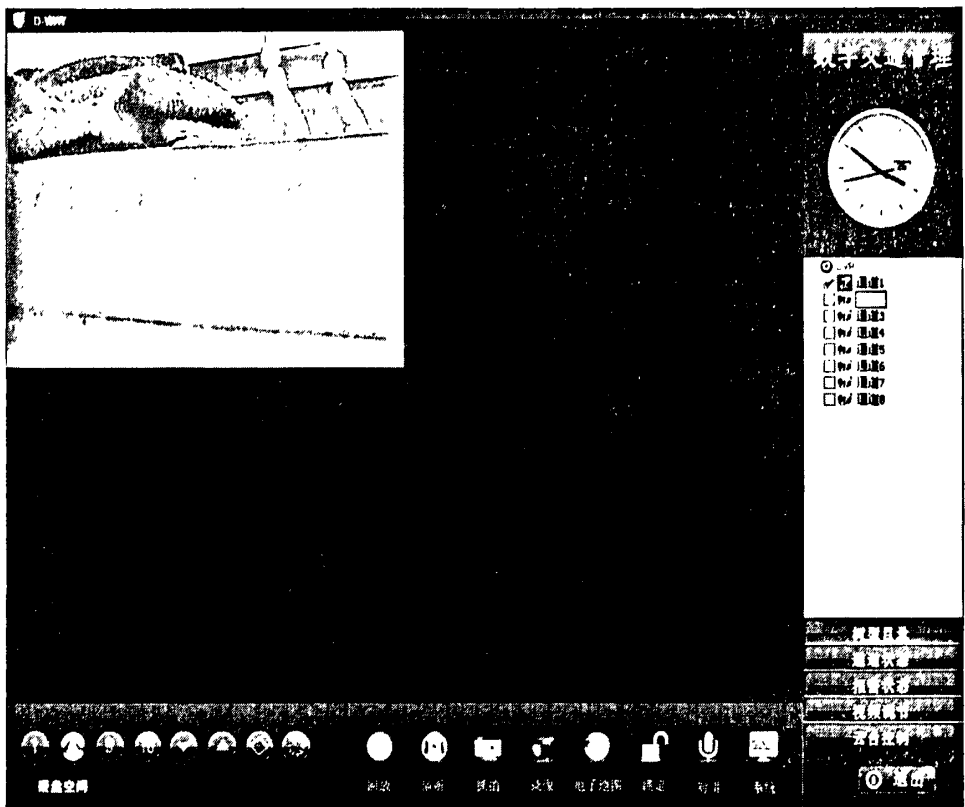


图4.4 预览界面

程序设定的逻辑是：如果某个资源被打开（无论它是区域、DVR或通道），那么它的孩子节点全部打开；如果同时其兄弟节点也已经打开，那么更新其父节点的状态为打开。

为此，在处理播放逻辑时设计了如下两个递归调用算法，如图4.5所示。

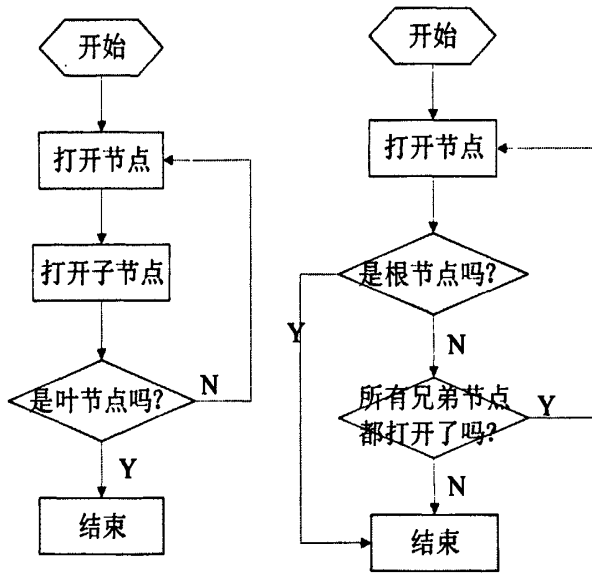


图4.5遍历资源树流程图

它们分别向上向下遍历整个资源树。假设当前节点为P:

(1) 检测P的所有兄弟节点，如果所有兄弟节点的状态都是“已播放”，则更新P的父节点的状态，然后继续按逻辑2递归处理P节点的父节点，直至树根结点；

(2) 向下遍历P的所有孩子节点，如果有未播放的通道则打开所有通道，同时更新状态为“已播放”，然后继续按逻辑1递归遍历P每个孩子节点，至根结点停止。

其中是视频参数设置界面如图4.6。

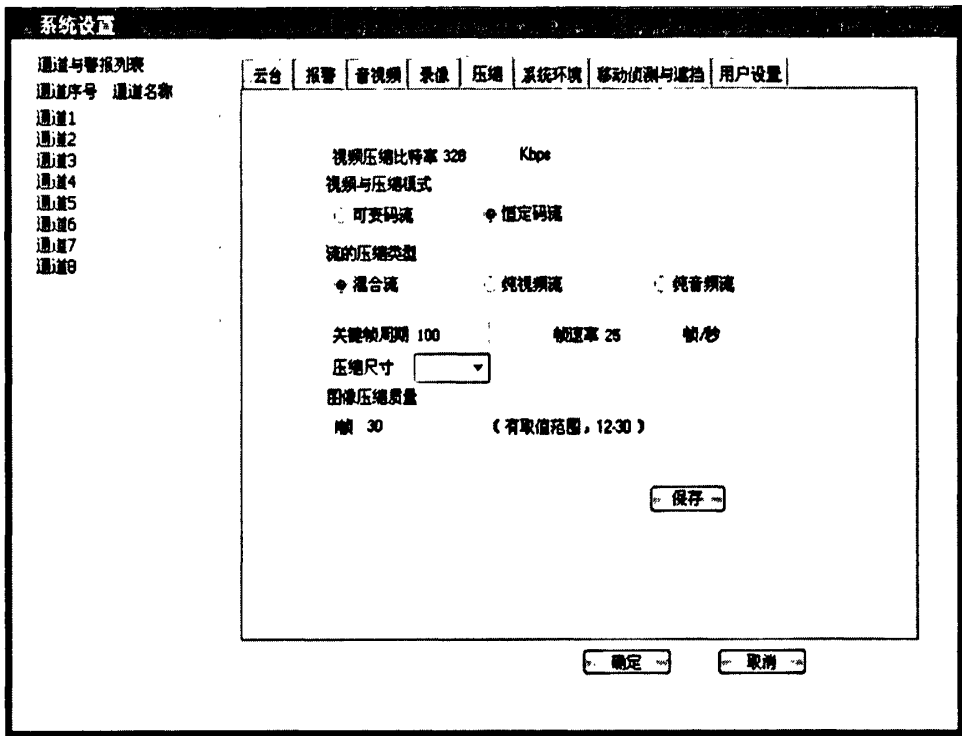


图4.6视频参数设置

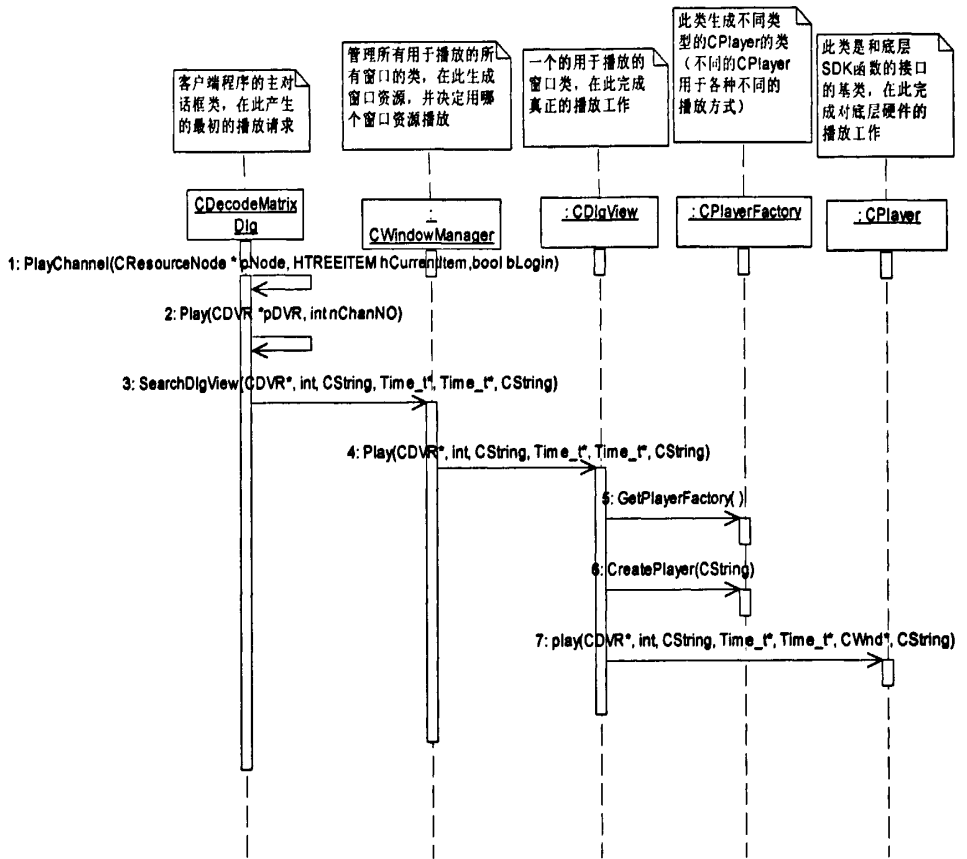


图4.7 播放逻辑

上图4.7是播放某个DVR通道的逻辑处理过程。当用户点击程序主窗口CDecodeMatrixDlg上的资源树节点（即通道）时，程序会根据该通道所属DVR的类型创建一个相应的播放类对象，然后找到（如果有的话）一个未播放的窗口播放。由于设计时采用了设计模式方法中的“工厂模式”，将嵌入式和PC式DVR间的硬件差别封装起来，对外采用统一的接口，降低了编程难度。

4.2.2 录像与回放

录像功能可由许多条件触发。最基本的是用户手动录像，其它还有自动录像和由于报警或移动侦测引起的触发录像。图4.8是录像类型示意图。

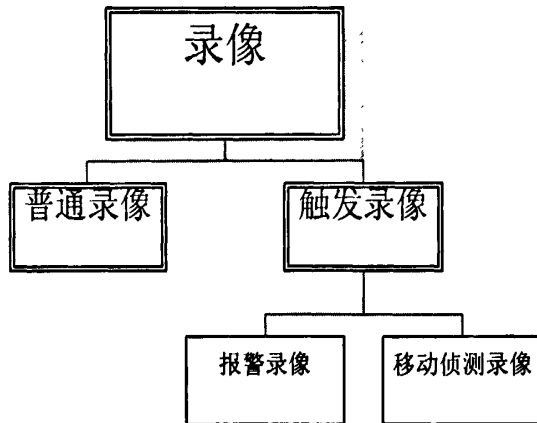


图4.8 录像结构图

具体的录像文件路径及文件名命名格式如下：

盘符（用户可以设置）:\mpeg4record\年-月-日\普通录像（或触发录像）\DVR的IP地址_通道号_年月日_时分秒.mp4

下面是一个具体的录像文件名及路径：

D:\mpeg4record\2010-04-11\普通录像\192.168.1.41_00_20100411_154326.mp4

点击“本地回放”确定后进入本地回放界面，如下：图中右上角的“打开文件”按钮用于打开本地的文件（只要知道录像文件的存放路径就行），再双击文件即可以回放。

右上角的“条件查询”按钮用于对本地录像文件的查找，点击按钮后进入如图界面，在弹出的对话框里面按自己的要求条件填写后，点击左侧树型目录下要查询的服务器，如图4.9。

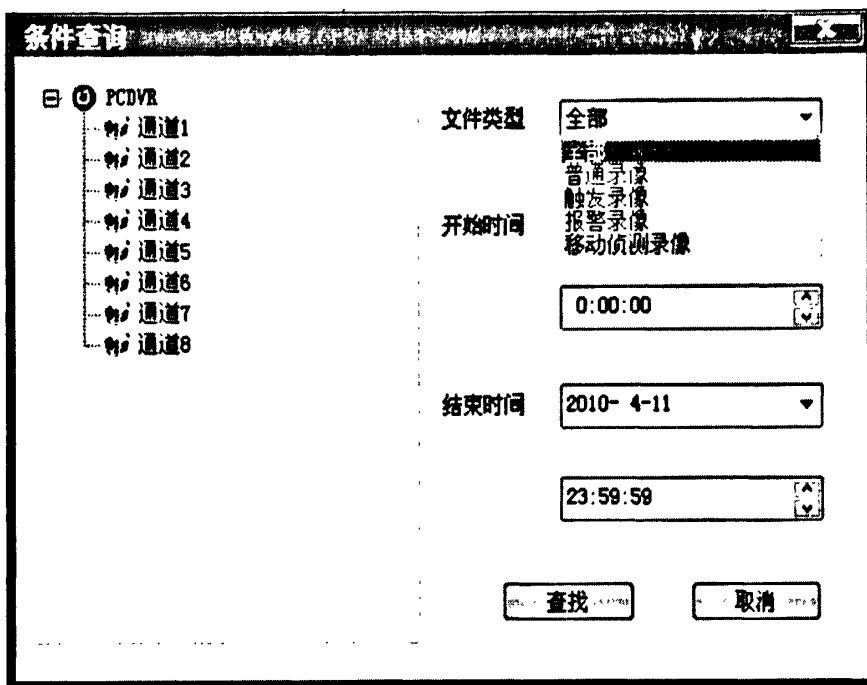


图4.9a 查询条件输入界面

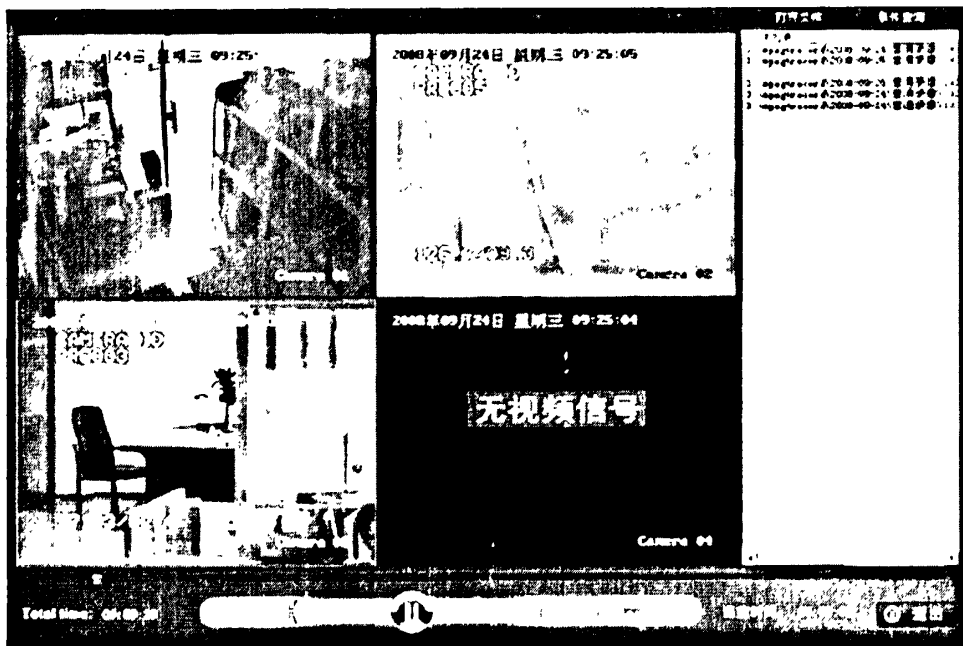


图4.9b 录像回放

点击确定后，在右侧的文件列表中会显示出符合条件的录像文件，双击要回放的录像文件即可以在左边的回放窗口中看到录像画面。

程序还集成了录像截取功能，用户可以设定所要截取录像的开始和结尾，就能从一个较长的录像文件中截取出想要的部分。

4.2.3 云台控制

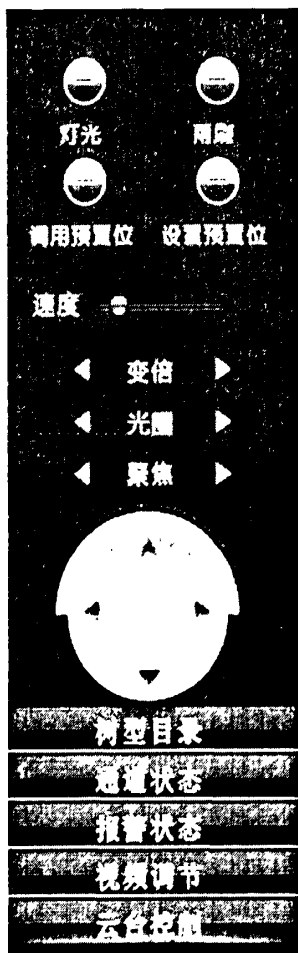


图4.10 云台控制界面

上图4.10的控制界面可以控制云台以步进方式向上、下、左、右转动，还能调节摄像头的变焦、光圈缩放及聚焦。

图4.11为云台参数设置界面。

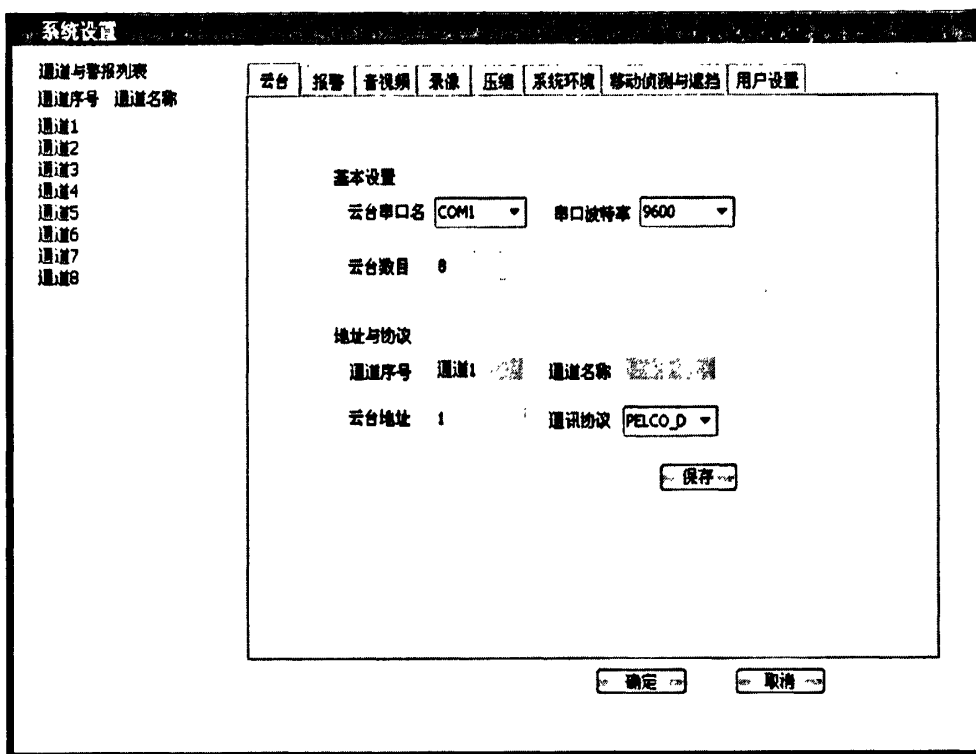


图4.11云台参数设置界面

云台通过PC机的串口与PCDVR相连，这里使用了云台控制协议PELCO-D和PELCO-P协议。将不同协议的处理过程封装在不同的类中，可以屏蔽了底层差别，并方便程序的编写。因为这两个协议类似，下面将以PELCO-P协议为例进行介绍：

PELCO-P协议的数据格式：1位起始位、8位数据、1位停止位，无校验位^[27]。
波特率：9600B/S

表4-1 命令格式

字节1	字节2	字节3	字节4	字节5	字节6	字节7	字节8
STX	地址码	指令码1	指令码2	数据码1	数据码2	ETX	校验码

说明：

- (1) 该协议中所有数值都为十六进制数
- (2) STX始终为A0H
- (3) 地址码为摄像机的逻辑地址号，地址范围：00H - 1FH
- (4) 指令码表示不同的动作
- (5) 数据码1、2分别表示水平、垂直方向速度（00-3FH），在有关预制点的

操作时，数据码2表示预制点值

(6) ETX始终为AFH

(7) 校验码(XOR sum of Bytes 2-6) = 字节2^字节3^字节4^字节5^字节6

以地址码0x01为例说明具体的命令：

{0xa0, 0x01, 0x00, 0x08, 0x00, 0x30, 0xaf, 0x39, } //上

{0xa0, 0x01, 0x00, 0x10, 0x00, 0x30, 0xaf, 0x21, } //下

{0xa0, 0x01, 0x00, 0x04, 0x10, 0x00, 0xaf, 0x15, } //左

{0xa0, 0x01, 0x00, 0x02, 0x10, 0x00, 0xaf, 0x13, } //右

{0xa0, 0x01, 0x00, 0x40, 0x00, 0x00, 0xaf, 0x41, } //变倍短

{0xa0, 0x01, 0x00, 0x20, 0x00, 0x00, 0xaf, 0x21, } //变倍长

{0xa0, 0x01, 0x02, 0x00, 0x00, 0x00, 0xaf, 0x03, } //聚焦近

{0xa0, 0x01, 0x01, 0x00, 0x00, 0x00, 0xaf, 0x00, } //聚焦远

{0xa0, 0x01, 0x08, 0x00, 0x00, 0x00, 0xaf, 0x09, } //光圈小

{0xa0, 0x01, 0x04, 0x00, 0x00, 0x00, 0xaf, 0x05, } //光圈大

无命令字发送//灯光关

无命令字发送//灯光开

{0xa0, 0x01, 0x00, 0x96, 0x00, 0x20, 0xaf, 0xb7, } //自动巡航

{0xa0, 0x01, 0x00, 0x99, 0x00, 0x20, 0xaf, 0xb8, } //关闭自动巡航

{0xa0, 0x01, 0x00, 0x07, 0x00, 0x01, 0xaf, 0x07, } //转至预置点001

{0xa0, 0x01, 0x00, 0x03, 0x00, 0x01, 0xaf, 0x03, } //设置预置点001

{0xa0, 0x01, 0x00, 0x05, 0x00, 0x01, 0xaf, 0x01, } //删除预置点001

以上对应的停命令均是（上下左右、变倍、聚焦、光圈均有停命令）：

{0xa0, 0x01, 0x00, 0x00, 0x00, 0x00, 0xaf, 0x01, }

4.2.4 移动侦测与遮挡

移动侦测这个功能主要是检测有无移动的物体。在设置移动侦测区域时以32*32像素块为单位，按4CIF（704*576）分辨率计算，每行有22个块（704/32），PAL时18行（576/32），NTSC时15行（480/32），与编码格式无关。经过测试，这种方法提高了灵敏度和可靠性，并简化了返回的数据，返回的值是18个DWORD，对应屏幕高度576/32=18行（PAL），每个DWORD的0-21位对应屏幕宽度704/32=22，其中1为运动，0为静止，也可以调用原有MotionAnalyzer分析结果。

硬件的能力是最多设置100个移动侦测框，而出于实际需要，每个通道最多设置4个就能满足要求。

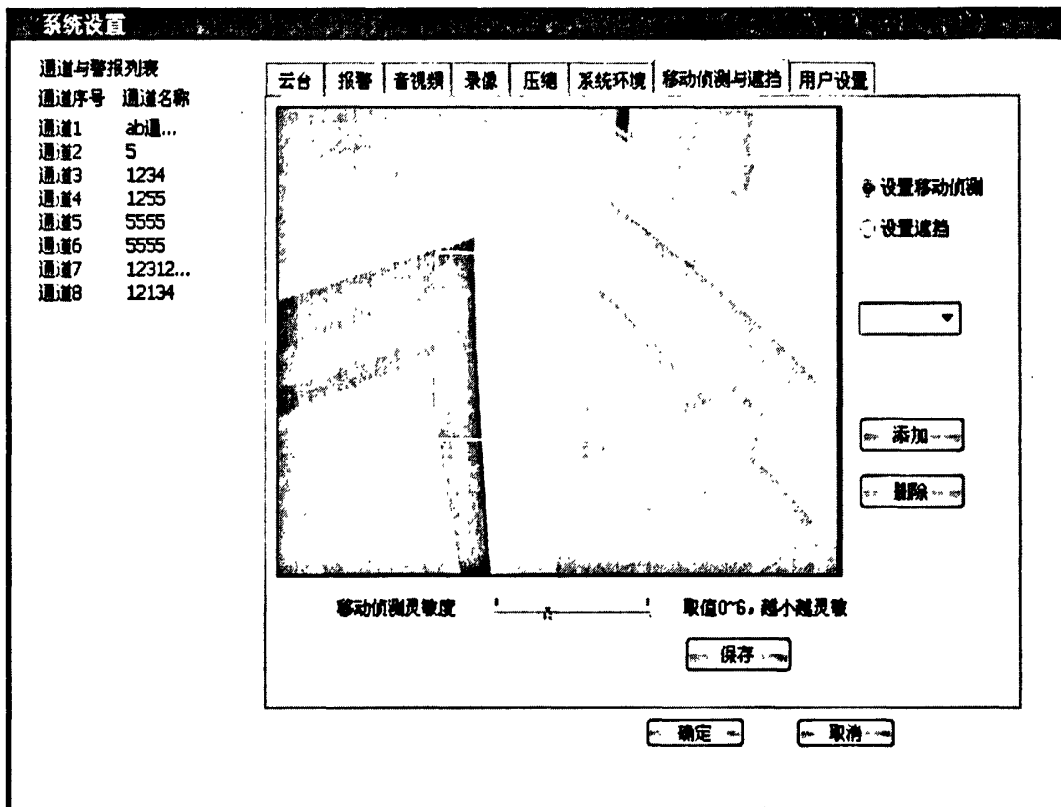


图4.12 移动侦测设置界面

遮挡就是用一块黑色方块覆盖设定的区域，防止某些隐私被摄入录像机。方案里最多支持四个遮挡区域。

有关移动侦测和遮挡的详细设计本文将在后面做详细说明。

4.2.5 电子地图

将通道和报警点以地图的形式显示在屏幕上，当通道或报警点发生报警事件，电子地图会自动弹出，并显示发生报警的区域地图，通道或报警点闪烁，双击通道可对前端动点进行控制。

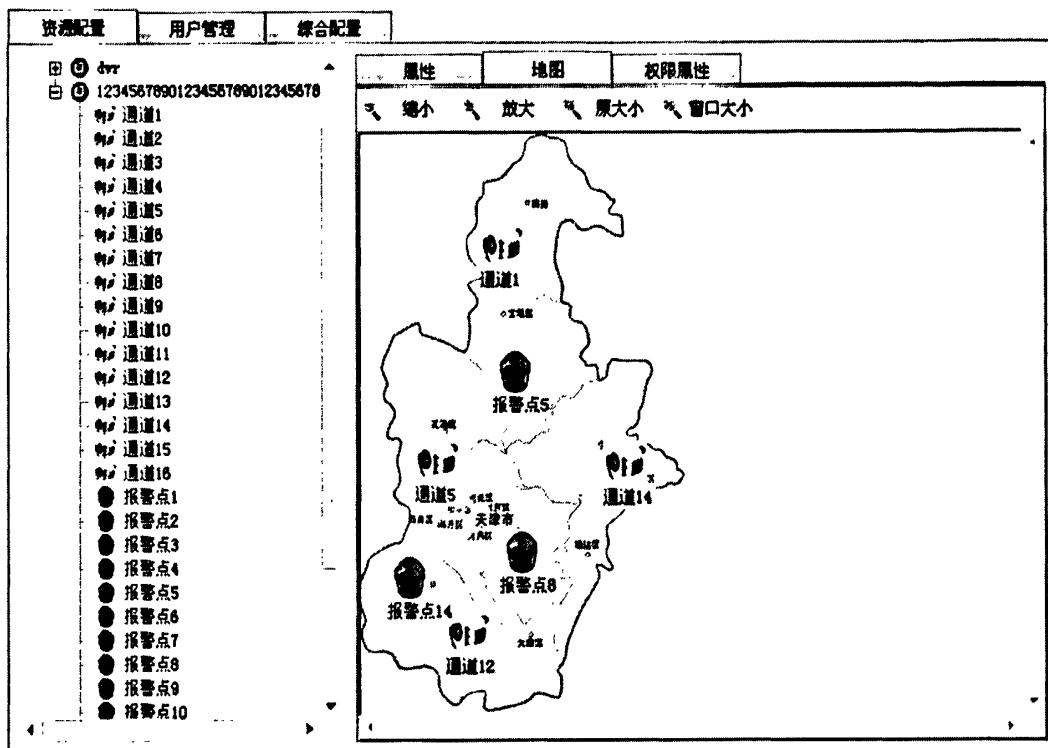


图4.13 报警设置界面

在实际运用中，PCDVR一般会连接A8报警箱进行报警的上传。下面讲述一下报警信息的上传。

如图4.13所示，如果发生报警，回调函数MessageCallBack（）就会立刻触发，然后调用CDVRMsgHandler中的报警信息处理函数，将报警信息按类型进行分类，最后发往上层界面CClientMainWnd类，显示给用户。

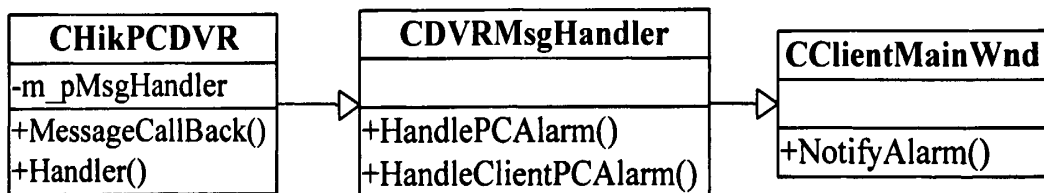


图4.14 报警上传流程

图4.14为报警参数设置界面。

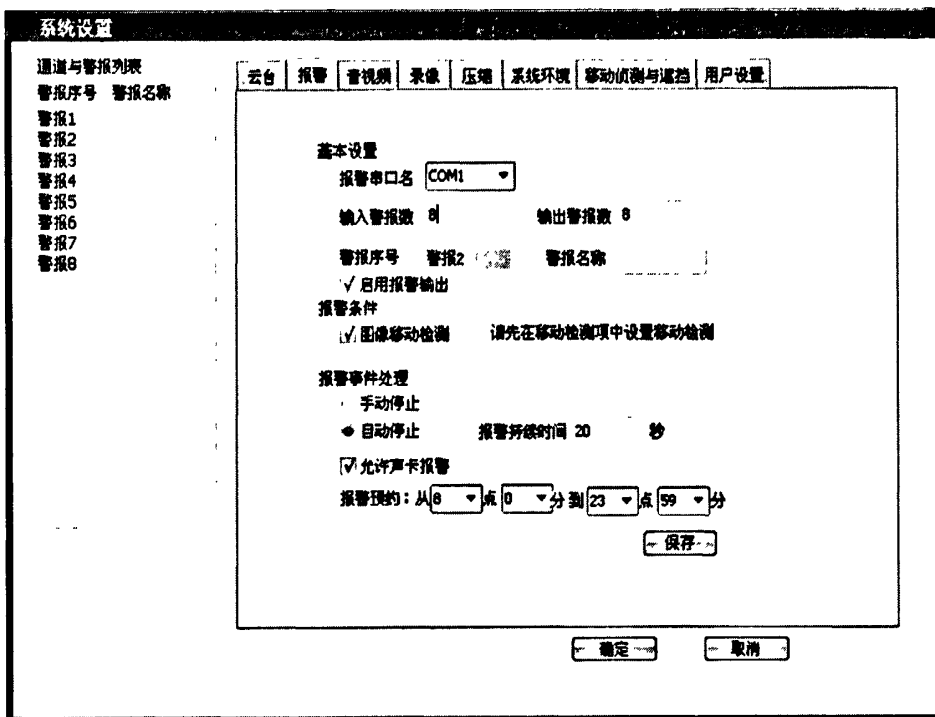


图4.15报警参数设置

图 4.16 为用户信息设置。

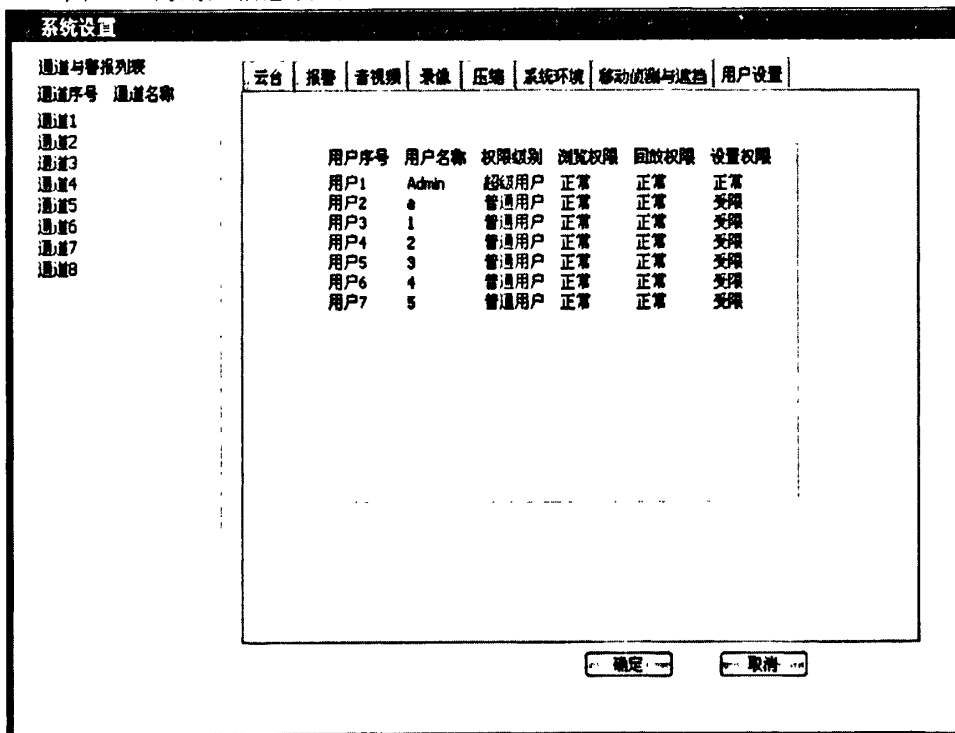


图4.16用户设置

第三节 技术难点及实现

4.3.1 配置信息的存储

系统要存储许多数据，它们分别控制不同功能。图 4.13 是系统的数据字典。

<table border="1"> <tr><th colspan="2">云台</th></tr> <tr><td> </td><td> </td></tr> <tr><td>波特率</td><td> </td></tr> <tr><td>协议</td><td> </td></tr> <tr><td>串口号</td><td> </td></tr> <tr><td>地址</td><td> </td></tr> <tr><td>云台数量</td><td> </td></tr> </table>	云台				波特率		协议		串口号		地址		云台数量		<table border="1"> <tr><th colspan="2">移动侦测</th></tr> <tr><th>PK</th><th>通道号</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>灵敏度</td></tr> <tr><td> </td><td>侦测区间1</td></tr> <tr><td> </td><td>侦测区间2</td></tr> <tr><td> </td><td>侦测区间3</td></tr> <tr><td> </td><td>侦测区间4</td></tr> </table>	移动侦测		PK	通道号				灵敏度		侦测区间1		侦测区间2		侦测区间3		侦测区间4	<table border="1"> <tr><th colspan="2">音视频设定</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>抓图保存地址</td></tr> <tr><td> </td><td>轮显时间间隔</td></tr> <tr><td> </td><td>是否打开预览</td></tr> <tr><td> </td><td>是否打开监听</td></tr> <tr><td> </td><td>自动锁定时间</td></tr> </table>	音视频设定					抓图保存地址		轮显时间间隔		是否打开预览		是否打开监听		自动锁定时间	<table border="1"> <tr><th colspan="2">用户</th></tr> <tr><th>PK</th><th>用户序号</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>用户名</td></tr> <tr><td> </td><td>用户密码</td></tr> <tr><td> </td><td>用户等级</td></tr> </table>	用户		PK	用户序号				用户名		用户密码		用户等级							
云台																																																																		
波特率																																																																		
协议																																																																		
串口号																																																																		
地址																																																																		
云台数量																																																																		
移动侦测																																																																		
PK	通道号																																																																	
	灵敏度																																																																	
	侦测区间1																																																																	
	侦测区间2																																																																	
	侦测区间3																																																																	
	侦测区间4																																																																	
音视频设定																																																																		
	抓图保存地址																																																																	
	轮显时间间隔																																																																	
	是否打开预览																																																																	
	是否打开监听																																																																	
	自动锁定时间																																																																	
用户																																																																		
PK	用户序号																																																																	
	用户名																																																																	
	用户密码																																																																	
	用户等级																																																																	
<table border="1"> <tr><th colspan="2">录像</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>打包时间</td></tr> <tr><td> </td><td>开始盘符</td></tr> <tr><td> </td><td>结束盘符</td></tr> <tr><td> </td><td>是否自动录像</td></tr> <tr><td> </td><td>是否循环录像</td></tr> <tr><td> </td><td>录像停止方式</td></tr> <tr><td> </td><td>自动录像条件</td></tr> <tr><td> </td><td>录像时间</td></tr> </table>	录像					打包时间		开始盘符		结束盘符		是否自动录像		是否循环录像		录像停止方式		自动录像条件		录像时间	<table border="1"> <tr><th colspan="2">视频压缩设定</th></tr> <tr><th>PK</th><th>通道号</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>图像压缩质量</td></tr> <tr><td> </td><td>压缩比特率</td></tr> <tr><td> </td><td>帧速率</td></tr> <tr><td> </td><td>关键帧周期</td></tr> <tr><td> </td><td>压缩尺寸</td></tr> <tr><td> </td><td>压缩模式</td></tr> <tr><td> </td><td>压缩范围</td></tr> </table>	视频压缩设定		PK	通道号				图像压缩质量		压缩比特率		帧速率		关键帧周期		压缩尺寸		压缩模式		压缩范围	<table border="1"> <tr><th colspan="2">设备</th></tr> <tr><th>PK</th><th>设备ID</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>设备名</td></tr> <tr><td> </td><td>设备地址</td></tr> <tr><td> </td><td>设备端口</td></tr> <tr><td> </td><td>通道数</td></tr> <tr><td> </td><td>用户名</td></tr> <tr><td> </td><td>用户密码</td></tr> <tr><td> </td><td>设备类型</td></tr> <tr><td> </td><td>报警数</td></tr> <tr><td> </td><td>电子地图名称</td></tr> </table>	设备		PK	设备ID				设备名		设备地址		设备端口		通道数		用户名		用户密码		设备类型		报警数		电子地图名称
录像																																																																		
	打包时间																																																																	
	开始盘符																																																																	
	结束盘符																																																																	
	是否自动录像																																																																	
	是否循环录像																																																																	
	录像停止方式																																																																	
	自动录像条件																																																																	
	录像时间																																																																	
视频压缩设定																																																																		
PK	通道号																																																																	
	图像压缩质量																																																																	
	压缩比特率																																																																	
	帧速率																																																																	
	关键帧周期																																																																	
	压缩尺寸																																																																	
	压缩模式																																																																	
	压缩范围																																																																	
设备																																																																		
PK	设备ID																																																																	
	设备名																																																																	
	设备地址																																																																	
	设备端口																																																																	
	通道数																																																																	
	用户名																																																																	
	用户密码																																																																	
	设备类型																																																																	
	报警数																																																																	
	电子地图名称																																																																	
<table border="1"> <tr><th colspan="2">遮挡</th></tr> <tr><th>PK</th><th>通道号</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>遮挡区间1</td></tr> <tr><td> </td><td>遮挡区间2</td></tr> <tr><td> </td><td>遮挡区间3</td></tr> <tr><td> </td><td>遮挡区间4</td></tr> </table>	遮挡		PK	通道号				遮挡区间1		遮挡区间2		遮挡区间3		遮挡区间4	<table border="1"> <tr><th colspan="2">网络设定</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>身份验证</td></tr> <tr><td> </td><td>是否允许回放</td></tr> <tr><td> </td><td>缓冲区大小</td></tr> <tr><td> </td><td>通信端口</td></tr> </table>	网络设定					身份验证		是否允许回放		缓冲区大小		通信端口	<table border="1"> <tr><th colspan="2">报警</th></tr> <tr><th>PK</th><th>报警序号</th></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td>报警输入数</td></tr> <tr><td> </td><td>报警输出数</td></tr> <tr><td> </td><td>串口名</td></tr> <tr><td> </td><td>报警持续时间</td></tr> <tr><td> </td><td>是否允许报警输出</td></tr> <tr><td> </td><td>是否允许移动侦测报警</td></tr> </table>	报警		PK	报警序号				报警输入数		报警输出数		串口名		报警持续时间		是否允许报警输出		是否允许移动侦测报警																				
遮挡																																																																		
PK	通道号																																																																	
	遮挡区间1																																																																	
	遮挡区间2																																																																	
	遮挡区间3																																																																	
	遮挡区间4																																																																	
网络设定																																																																		
	身份验证																																																																	
	是否允许回放																																																																	
	缓冲区大小																																																																	
	通信端口																																																																	
报警																																																																		
PK	报警序号																																																																	
	报警输入数																																																																	
	报警输出数																																																																	
	串口名																																																																	
	报警持续时间																																																																	
	是否允许报警输出																																																																	
	是否允许移动侦测报警																																																																	

图4.13 数据字典

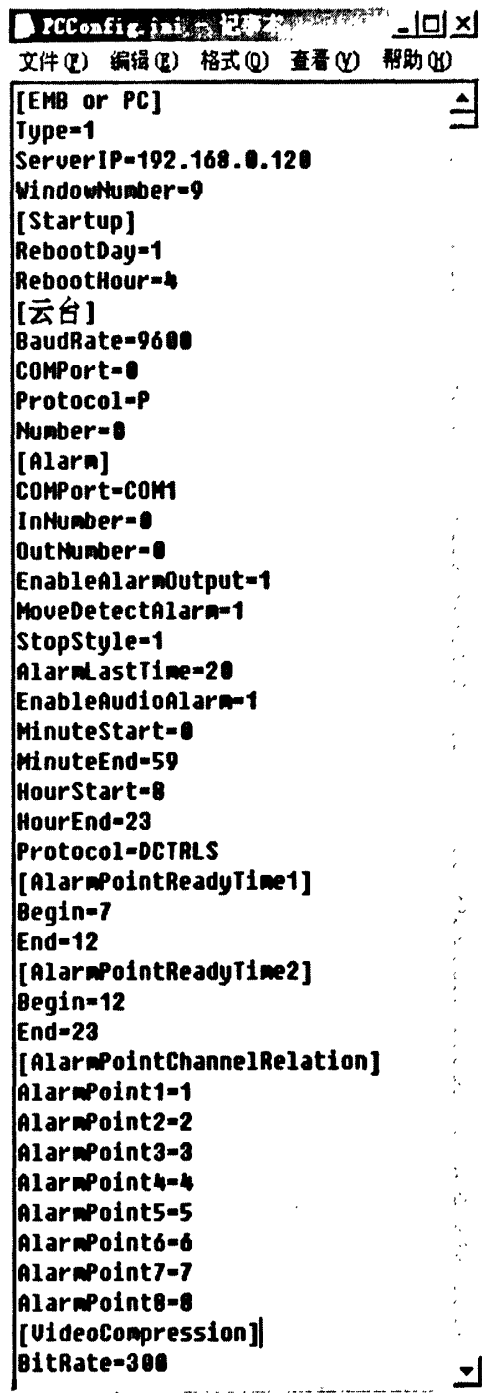
这些数据主要是存储在本地的PCDVR上，虽然每一项所占空间不大，但数量很多（移动侦测、遮挡、报警这三种设定是与通道相关的，通道越多设定就越

多), 所以存储所需的工作量还是很大的。

考虑到这些实际情况, 这里用配置文件“*.ini”的存储方式来存储这些设定参数。

下面简单介绍一下配置文件的存储方式: 配置文件的分隔单位是字段和键, 一个配置文件下可以有多个字段, 字段名不能重复; 而每个字段下又可以有多个键, 同一字段键名不能重复, 不同字段可以有相同的键名。在实际使用过程中, 可以通过读取和改写某个字段下的某个键名来起到储存信息的目的。

图 4.14 是系统配置文件的截图。



```
[EMB or PC]
Type=1
ServerIP=192.168.0.120
WindowNumber=9
[Startup]
RebootDay=1
RebootHour=4
[云台]
BaudRate=9600
COMPort=0
Protocol=P
Number=0
[Alarm]
COMPort=COM1
InNumber=0
OutNumber=0
EnableAlarmOutput=1
MoveDetectAlarm=1
StopStyle=1
AlarmLastTime=20
EnableAudioAlarm=1
MinuteStart=0
MinuteEnd=59
HourStart=8
HourEnd=23
Protocol=DCTRLS
[AlarmPointReadyTime1]
Begin=7
End=12
[AlarmPointReadyTime2]
Begin=12
End=23
[AlarmPointChannelRelation]
AlarmPoint1=1
AlarmPoint2=2
AlarmPoint3=3
AlarmPoint4=4
AlarmPoint5=5
AlarmPoint6=6
AlarmPoint7=7
AlarmPoint8=8
[VideoCompression]
BitRate=300
```

图4.14 配置文件

使用上面介绍的配置信息存储方法后，系统就能够方便快捷地存取本地的配置信息，而且能做到设置改变即能生效，不需重启程序，同时也大大减轻了

编程负担。

图4.15是PCDVR本地录像参数设置界面图。

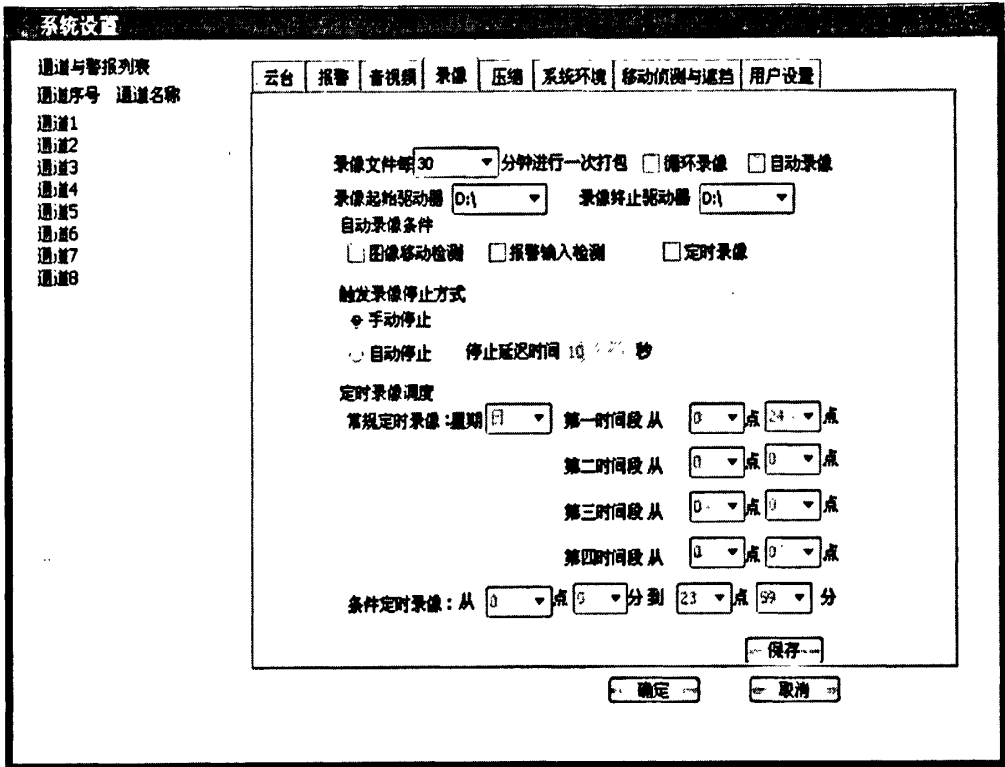


图4.15 录像参数设置界面图

这个界面可以设定录像参数：如打包时间、是否循环录像、是否自动录像、录像起始及终止驱动器、自动录像触发条件、触发录像停止方式、定时录像时间段设置（一周七天、每天四个时间段）、条件定时录像时间段。

4.3.2 性能优化

1、录像删除

程序之前的录像删除策略是每隔几分钟查看录像盘符的剩余空间，如果空间低于设定值则删除时间最早的那个录像，但在实际应用过程中发现由于某些特殊（但正常）的原因，某些录像所占空间可能比较小，而在正常状态下一个通道一小时能够记录 100M（或者更多）的数据，当录像通道有十几路的话对磁盘的消耗是非常大的，如果删除文件速度过慢会经常导致磁盘被写满，导致程序错误。后来经过大家讨论，决定每次清理时删除最早的一天的录像，这样每次能清理出几个 G 的空间，基本不会发生磁盘满的现象，而且使得轮巡检测的

函数也不需运行得太频繁，降低了系统负荷。

2、提高系统运行效率

虽然系统比较复杂，实现的功能较多，但是占用资源、CPU 使用率却不能太高，同时，考虑到人机交互，为了给用户一个方便快捷的操作界面，系统的反应速度不能低。在提高系统运行效率方面做了如下工作：

(1) 将一些处理时间长、系统负担重、不要求立刻返回结果的操作开辟出另一个线程来进行后台处理，这样可以防止程序界面假死；

(2) 出于系统功能的角度考虑，系统最初使用了许多定时器来进行定时功能，它们增加了系统的负担，造成定时不准的状况。在软件后期系统测试维护阶段去掉了不必要的定时器，使得系统能够正常高效地执行；

(3) 对一些关键操作的逻辑处理做了优化（例如改变预览窗口个数），使得系统运行速度更快。

4.3.3 移动侦测、遮挡功能中区域位置的存储

系统每个通道能够存储四个移动侦测和四个遮挡区域，由于预览窗口能够控制放大缩小，为了使窗口大小的变化不影响这些特定区域相对于整个窗口的位置，对区域位置的存储做了特别设计。

图 4.16 是计算区域相对于整个窗口位置比例的示意图。

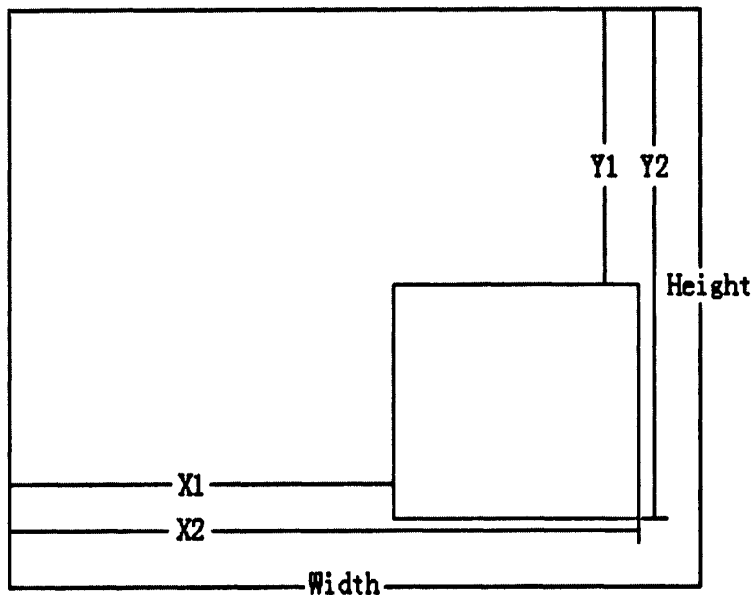


图4.16 窗口比例计算

分别计算了区域上下左右四个边界分别占整个窗口高和宽的比例，然后将这个比例存下来，公式如下：

$$\frac{X_1}{Width}, \frac{X_2}{Width}, \frac{Y_1}{Height}, \frac{Y_2}{Height} \quad (4.1)$$

这样，无论窗口大小如何变化，区域的相对位置都不变。

4.3.4 录像远程查询和下载

1、在监控系统在实际使用中，用户经常会在客户端远程操作PCDVR的录像，因此就需要录像远程查询功能。

远程录像搜索的逻辑类似于本地录像搜索，查询条件包括查询的时间范围（年月日时分）、录像类型、DVR信息。程序采用Windows Socket编程方法在PCDVR和客户端程序之间建立Socket连接，客户端向PCDVR发送查找命令，PCDVR在本地搜索完毕后将搜索结果通过局域网上的Socket连接发送给查询的客户端。

录像远程查询这个功能对编程人员的要求是很高的，因为需要考虑到很多特殊情况，比如多人同时查询一台PCDVR的录像信息、因为网络原因导致的连接中断等问题。

为此，程序中加入了一些特殊的技术手段。监听事先约定好的端口，当发现有客户端发来的查询命令后，首先建立与该客户端的Socket连接，新建一个线程查找本地录像文件，然后将结果通过该连接返回给客户端，然后中断连接。考虑到PCDVR在录像外剩余的执行能力及搜索算法对设备的负担，程序设定每个PCDVR最多建立3条连接。

同时，考虑到网络可能会发生的问题，程序设定，如果客户端与PCDVR的连接如果非正常中断30秒以上，则释放Socket占用的资源。

2、在远程回放PCDVR的录像后想将其中的某段录像下载到本地供后续使用，因此系统需要集成远程下载功能。

对于远程下载功能有如下性能要求：

(1) 传输准确性高：由于视频监控系统均是用在一些重要场合，如银行、超市、交通路口等，因此录像中的信息都是十分重要的，如果在远程传输中有任何信息的丢失，这个责任谁也负不起的。因此，下载的正确性必须做到100%。

(2) 下载速度也是远程下载的一项重要指标。如果用户在下载一个100M左右

的录像文件时能够有时间喝一杯咖啡，那么没有一个人会选择这款监控系统。因此，在保证正确性的前提下，程序要尽量提高远程下载速度。

在编写录像远程查询和下载功能的时候，为了能够传输所需的控制命令，程序中新建了两个无模式对话框。无模式对话框也是一种没有界面的对话框，可以利用它来传递消息，但它对开发者是隐形的，对用户是透明的。

第四节 软件测试与软件维护

4.4.1 软件测试

在软件开发的一系列活动中，为了保证软件的可靠性，人们研究并使用了很多方法进行分析、设计及编码实现。但是由于软件产品本身是无形的、复杂的、知识密集的逻辑产品，其中难免有各种各样的错误，因此，需要通过测试查找错误，保证软件的质量。软件测试是保证软件质量的关键，它是对需求分析、设计和编码的最终复审^[28]。

软件测试的原则：

- (1) 测试用例既要有输入数据，又要有对应结果。这样便于对照检查，做到“有的放矢”。
- (2) 测试用例不仅要选用合理的输入数据，还应选择不合理的输入数据。这样能更多地发现错误，提高程序的可靠性，还可以测试出程序的排错能力。
- (3) 除了检查程序是否做了它应该做的工作，还应该检查程序是否做了它不应该做的工作。
- (4) 应该远在测试开始之前就制定测试计划。实际上，一旦完成了需求分析模型就可以开始制定测试计划，在建立了设计模型之后就可以立即开始设计详细的测试方案。因此，在编码之前就可以对所有测试工作进行计划和设计，并严格执行，排除随意性。
- (5) 测试计划、测试用例、测试报告必须作为文档长期保存。因为程序修改以后有时可能会引进新的错误，需要进行回归测试。同时可以为以后的维护提供方便，对新人或今后的工作都有指导意义。
- (6) Pare to原理说明，测试发现的错误中80%很可能是由程序中20%的模块造成的，即错误出现的“群集性”现象。

(7) 为了达到最佳的测试效果,程序员应该避免测试自己的程序。测试是一种“挑剔性”的行为,测试自己的程序存在心里障碍。另外,对需求规格说明的理解而引入的错误则更不容易发现。因此,应该由独立的第三方从事测试工作会更客观、更有效。

测试是一个执行程序的过程。软件测试的方法一般分为动态测试和静态测试。把被测程序在机器上运行称为动态测试,不在机器上运行被测程序称为静态测试。

静态测试就是静态分析,它采用的手段有两种:人工检测和计算机辅助静态分析。

(1) 人工测试指不依靠计算机而完全靠人工审查程序或评价软件。人工审查程序偏重于编码风格、编码质量的检验,除了审查编码,还要对各阶段的软件产品进行检验。人工测试可以有效地发现软件的逻辑设计和编码错误,发现计算机不容易发现的错误。

(2) 计算机辅助静态分析是指利用静态分析工具对被测程序进行特征分析,从程序中提取一些信息,以便检查程序逻辑的各种缺陷和可疑的程序构造。

动态测试是指通过运行程序发现错误,方法也有两种:黑盒测试法和白盒测试法。

黑盒测试也称功能测试或数据驱动测试。它不考虑程序内部结构和处理过程。把被测程序看成一个黑盒子,只在软件接口处进行测试,依据需求规格说明书,检查程序是否满足功能要求。每个功能是否都能正常使用,是否满足用户的要求,程序是否能适当地接受输入数据并产生正确的输出信息,并且保持外部信息的完整性。

通过黑盒测试主要发现以下错误:

- (1) 是否有不正确或遗漏了的功能;
- (2) 在接口上,能否正确地接受输入数据,能否产生正确的输出信息。
- (3) 访问外部信息是否有错。
- (4) 性能上是否满足要求。
- (5) 界面是否有错,是否美观、友好。

白盒测试也称结构测试或逻辑驱动测试。白盒测试中,测试人员需了解程序的内部结构和处理过程,以检查处理过程的细节为基础,要求对程序的结构特性做到一定程序的覆盖,对程序中的所有逻辑路径进行测试,并检验内部控

制结构是否有错，确定实际的运行状态与预期的状态是否一致。

图4.17是一个典型的软件测试过程示意图。

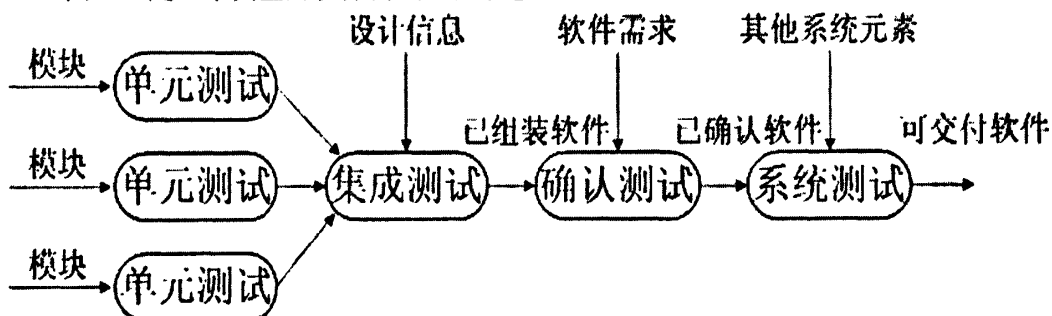


图4.17 软件测试过程

监控系统一旦架设起来开始运行，就要求软件不能出错、不能停止。因此对软件的安全性、稳定性要求非常高。为了达到用户的要求，在系统后期测试过程中使用工具BoundsChecker进行计算机辅助静态分析测试。

BoundsChecker是Compuware公司开发的一款产品，这个软件属于Compuware的Numega系列（自动化的白盒测试工具）。BoundsChecker提供清晰的、详细的程序错误分析，自动查明静态的堆栈错误及内存/资源泄露，并能够迅速的定位出错的源代码，即使在没有源代码的情况下也可检查第三方组件的错误^[31]。

错误检测范围主要包括如下：

1、指针和泄露错误：

- (1) 接口泄露；
- (2) 内存泄露；
- (3) 资源泄露；
- (4) 未分配的指针错误；

2、内存错误：

- (1) 动态存储溢出；
- (2) 无效的句柄被锁定；
- (3) 句柄没有被锁定；
- (4) 内存分配冲突；
- (5) 栈空间溢出；
- (6) 静态存储溢出；

3、API和OLE错误：

- (1) API函数返回失败;
- (2) API函数未执行;
- (3) 无效的变量(包括指针变量、字符串变量等);
- (4) OLE接口方法的变量无效;
- (5) OLE接口方法失败;
- (6) 线程调用库函数错误;

4.4.2 软件维护

传统上,软件系统交付之后对其实施更改的学科叫做软件维护。通俗地说,软件维护是指软件系统交付使用以后,为了改正软件运行错误,或者因满足新的需求而加入新功能的修改软件的过程。

软件系统开发完成交付用户使用后,就进入软件的运行和维护阶段。软件维护阶段是软件生存周期中时间最长的一个阶段,所花费的精力和费用也是最多的一个阶段。有关软件系统交付之后对系统实施更改的特点和成本调查结果说明,软件系统整个生存周期总成本大约40%~70%要用于软件维护。软件系统经常会发生一些变化,如对隐含错误的修改、新功能的加入、环境变化造成的程序变动等。因此,需要充分认识软件维护工作的重要性和迫切性,以提高软件的可维护性,尽量减少软件维护的工作量和费用,从而提高软件系统的整体效益^[35]。

软件维护活动可归结为四类:

(1) 改正性维护:软件测试不可能找出一个软件系统中所有潜伏的错误,所以软件在交付使用后,必然会有一部分隐藏的错误被带到运行阶段中。这些隐藏的错误在某些特定的使用环境下才会暴露出来。把在软件投入使用后才逐渐暴露出来的错误的诊断、定位、改错的过程,称为改正性维护。

改正性维护中的主要维护策略有:开发过程中采用新技术、利用应用软件包、提高系统结构化程度、进行周期性维护审查等。

(2) 适应性维护:适应性维护是为了适应计算机的飞速发展,使软件适应外部新的硬件和软件环境或者数据环境发生的变化,而进行修改软件的过程。

适应性维护中的主要维护策略有:对可能变化的因素进行配置管理、将因环境变化而必须修改的部分局部化,即局限于某些程序模块等。

(3) 完善性维护:在软件漫长的使用过程中,为了满足用户使用过程中对软

件提出的新的功能与性能要求，需要对原来的软件的功能进行修改或扩充。这种为扩充软件功能、增强软件性能、提高软件运行效率和可维护性而进行的维护活动称为完善性维护。

完善性维护中的主要维护策略有：尽量采用功能强、使用方便的工具，采用原型化的开发方法等。

(4) 预防性维护：预防性维护是为了提高软件未来的可维护性、可靠性等，或为了给未来的改进奠定更好的基础而修改软件的过程。通常，预防性维护定义为：“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，该类维护工作需要采用先进的软件工程方法，对需要维护的软件或软件中的某一部分进行设计、编码和测试。

预防性维护中的主要维护策略是：采用提前实现、软件重用等技术。

软件维护中存在的问题：

(1) 读懂别人的程序是很困难的，而文档的不足更增加了这种难度。一般开发人员都有这样的体会，修改别人的程序还不如自己重新编写程序。

(2) 文档的不一致性是软件维护困难的又一个因素，主要表现在各种文档之间的不一致以及文档与程序之间的不一致性，从而导致维护人员不知所措，不知怎样进行修改。这种不一致性是由于开发过程中文档管理不严造成的，开发中经常会出现修改程序而忘了修改相关的文档，或者某一个文档修改了，却没有修改与之相关的其他文档等现象，解决文档不一致性的方法就是要加强开发工作中文档的版本管理。

(3) 软件开发和软件维护在人员和时间上存在差异。如果软件维护工作是由该软件的开发人员完成，则维护工作相对比较容易，因为这些人员熟悉软件的功能和结构等。但是，通常开发人员和维护人员是不同的，况且维护阶段持续时间很长。

(4) 软件维护不是一件吸引人的工作。由于维护工作的困难性，维护经常遭受挫折，而且很难出成果，所以高水平的程序员自然不愿主动去做，而公司也舍不得让高水平的程序员去做。

在开发PCDVR的同时也需要维护嵌入式DVR的版本，

这套解决方案在许多地方已经投入商业运营，例如学校（浙江某医学院）、交通系统（某城市市内交通监控）、银行（天津市某行多个区属分行）、矿山（山西某煤矿安全监控）小区（某城市开发区数字城管监控系统）。这些项目由于其

自身特点对监控系统的某项功能有其独特要求，有的项目对电视墙的功能有较高要求，有的项目则对录像、报警的要求比较高；同时用户又希望把不会用到的功能尽量屏蔽，避免误操作。虽然有些功能的改动看起来很简单，但对编程人员来说是一项艰苦的任务。

第五章 总结与展望

总结

本课题的研究是建立在实际课题的基础上, 本文结合当今网络视频监控系统的发展方向, 考虑了实际中遇到的各种问题, 运用面向对象程序设计、软件复用、设计模式、协作开发、软件测试等技术成功实现了既定功能。为了增加软件的易维护性、可扩展性和易用性, 本文采用快速原型法的开发模式; 为了兼容多家厂商的硬件设备, 采用了多层架构, 屏蔽了底层硬件的差异, 对用户做到透明。

本文的主要工作体现在以下几个方面:

(1) 以软件工程整个生命周期为主线, 分别详细介绍了系统的需求分析、概要设计、详细设计、测试、维护等关键环节, 清晰地展示了软件整个编写流程, 方便读者的阅读。

(2) 运用面向对象技术, 引入了多种设计模式模型、软件复用、软件重构等软件开发技术加强了模块内部的聚合性, 降低了模块间的耦合性, 提高了系统设计和开发的效率。

(3) 从系统实际需要出发, 针对方案开展的具体硬件环境和要完成任务的特殊性, 对软件开发做特别安排, 使设计更加合理, 开发过程更加顺利。

展望

未来, 视频监控系统会向前端一体化、视频数字化、监控网络化、系统集成化方向发展, 对系统的要求也会越来越高。后面还有许多工作要做。

(1) 随着监控设备硬件的发展, 图像清晰度越来越高, 存储所需空间也越来越大, 监控系统将来需要支持超大存储容量的磁盘阵列。

(2) 在一些对电子地图功能要求高的监控环境中(如城市交通管理系统、城市供暖管理系统等), 由于监控点分布范围较广, 用户希望地图放大后能够保证显示同样清晰。接下来准备开发的支持GIS系统的电子地图能够满足上述要求。

(3) 可进行人的行为分析(包括汽车拍照检测)的监控功能也在开发计划中。

(4) 可控制多个云台的云台矩阵、用于录像存储和回放的集中存储服务器、

流媒体服务器也在开发计划中。

面向对象的软件设计开发的过程比较复杂，但对增加开发人员的开发经验有很大帮助。相信随着科学技术的发展，面向对象软件技术会越来越地结合其他软件开发方法，使软件开发过程更加灵活、快捷、高效！

参考文献

- [1] 张海藩著. 软件工程. 北京: 人民邮电出版社, 2006
- [2] 李伟波, 王庆春, 刘永祥著. 软件工程学习与实践. 武汉: 武汉大学出版社, 2006
- [3] 视频监控系统的技术发展阶段: <http://www.secu.com.cn/news/view/200412/245825.shtml>
- [4] 软件工程 (百度百科): http://baike.baidu.com/view/1659.htm?fr=ala0_1_1#11
- [5] 谷千书. 软件工程系统结构模型的应用分析: [硕士学位论文]. 上海: 复旦大学, 2008
- [6] Ian Sommerville著, 程成, 陈霞译. 软件工程. 北京: 机械工业出版社, 2007
- [7] 李东生, 崔冬华, 李爱萍著. 软件工程原理、方法和工具. 北京: 机械工业出版社, 2009
- [8] 设计模式 (百度百科): <http://baike.baidu.com/view/66964.htm#2#2>
- [9] 林常须. 多客户端设计模式的研究与应用: [硕士学位论文]. 兰州: 兰州理工大学, 2005
- [10] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides 等著, 李英军等译. 设计模式: 可复用面向对象软件的基础 (Design Patterns Elements of Reusable Object-Oriented Software), 机械工业出版社, Addison-Wesley, 2000.9, 2~3, 121~128
- [11] Erich Gamma, Richard Helm, Ralph Johnson etc. Design Patterns: Elements of Reusable Object-Oriented Software. USA: Addison2Wesley, 1995, 14~30
- [12] Dong Jing, Yang Sheng, Zhang Kang. VisDP: a Web service for visualizing design patterns on demand[C]. In Proceedings of international conference. Information Technology: Coding and Computing(ITCC'05), April, 2005, 389~391
- [13] 严悍, 刘冬梅, 赵学龙著, UML软件建模、概念、规范与方法. 北京: 国防工业出版社, 2009
- [14] 黄远航. 基于构件模型和UML建模的信息系统研究与应用: [硕士学位论文]. 广州: 广东工业大学, 2005
- [15] 张鸷. 网络视频监控软件系统的设计与实现: [硕士学位论文]. 大连: 大连理工大学, 2009
- [16] 袁志勇, 王景存著. 嵌入式系统原理与应用技术. 北京: 航空航天大学出版社, 2009
- [17] 方帅. 计算机智能视频监控系统关键技术研究: [博士学位论文]. 沈阳: 东北大学, 2005
- [18] 何华丽. 基于H264的多平台视频监控系统的研究与实现: [硕士学位论文]. 北京: 北京邮电大学, 2009
- [19] 莫勇腾著. 深入浅出设计模式. 北京: 清华大学出版社, 2006
- [20] 于雷. 面向对象的软件开发方法在网络视频监控系统中的应用: [硕士学位论文]. 天津: 南开大学, 2008
- [21] Leszek A. Maciaszek著, 马素霞, 王素琴, 谢萍等译. 需求分析与系统设计. 北京: 机械工业出版社, 2009
- [22] Stephen P.Berczuk, Software Configuration Management Patterns Effective Teamwork, Practical Integration, Beijing, China electric Power Press, 2004, 1~94
- [23] Toby Segaran 著. Programming collective intelligence. 南京: 东南大学出版社, 2008

- [24] Len Bass, Paul Clements, Rick Kazman, Mark Klein. Evaluating the Software Architecture Competence of Organizations. Boston: Addison Wesley Professional, 2003
- [25] IEEE. Recommended Practice for Architectural Description of Software Intensive Systems. IEEE, IEEE Std1471-2000, 2000
- [26] 刘炳均. 面向对象软件的设计开发方法与配置管理研究: [硕士学位论文]. 天津: 南开大学, 2008
- [27] 云台控制协议: <http://www.qianjia.com/html/2007-03/24976.html>
- [28] 软件测试 (百度百科): <http://baike.baidu.com/view/16563.html?wtp=tt#11>
- [29] Glenford J.Myers 著, 周之英, 郑人杰译. 计算机软件测试技巧. 北京: 清华大学出版社, 2008
- [30] Stephen H. Kan 著, 吴明晖, 应晶等译. 软件质量工程 度量与模型 (第二版). 北京: 电子工业出版社, 2009
- [31] BoundsChecker 简介: <http://www.fztesting.com/viewthread.php?tid=147>
- [32] Danijel Arsenovski 著, 冯飞译. 代码重构. 北京: 清华大学出版社, 2009
- [33] Marilyn Bohl, Maria Rynn 著, 张光庭, 马国富, 王崧译. 结构化与面向对象程序设计. 北京: 电子工业出版社, 2008
- [34] 叶乃文, 王丹著, 面向对象程序设计, 北京 清华大学出版社, 2009
- [35] William F.Tuohey, Benefits and Effective Application of Software Engineering Standards, Software Quality Journal, 2002.10, 47~68
- [36] Russell J.T.Dyer著, 李红军, 李冬梅等译. MYSQL核心技术手册. 北京: 机械工业出版社, 2009
- [37] Michael J.Donahoo, Kenneth L.Calvert著, 陈宗斌等译. TCP/IP Sockets编程——C语言实现. 北京: 清华大学出版社, 2009
- [38] 朱娜敏, 魏宗寿, 李红编著. 精通Windows程序设计——基于Visual C++实现. 北京: 人民邮电出版社, 2009
- [39] 李媛媛编著. Visual C++网络通信开发入门与编程实践. 北京: 电子工业出版社, 2008
- [40] 刘元, 卢少同, 周亮. 网络视频监控系统的设计与实现. 网络与通信, 2009 (25): 129~131
- [41] 孟旭霞, 覃少华, 王艳妹. 嵌入式远程视频监控系统的设计与实现. 嵌入式系统应用, 2009 (25): 38~40
- [42] 尧敏. 基于.NET 架构的网络视频监控系统的设计与实现: [硕士学位论文]. 武汉: 华东交通大学, 2009
- [43] 张明杰. 远程视频监控系统中网络通信模块的设计. 工矿自动化, 2009 (1): 88~90
- [44] 罗尧, 贺武林, 汪厚祥. 多级网络视频监控系统的设计与实现. 计算机与数字工程, 2009 (37): 65~68

致 谢

时间过得好快，转眼间我的学生时代就要结束了，心里还真有几分舍不得。

回想起我 25 年的人生经历，我要首先感谢我的父母，他们含辛茹苦地把我养大，教我如何为人处世，自己省吃俭用供我读书学习。没有他们的勤劳简朴，没有他们对我的无微不至的爱，就没有我的今天。爸爸妈妈，谢谢你们！以后儿子会好好孝顺你们的！祝老爸老妈身体健康！天天快乐！

我要感谢陈增强、刘忠信二位老师。我的本科也是在南开渡过的，二位老师从本科起就关心帮助我的学习生活，研究生阶段更是如此。二位老师对我的谆谆教诲、悉心指导我会永远记在心里！祝二位老师身体健康、全家幸福！

感谢刘炳均、于雷两位师兄细心、无私地指导，在你们身上学到的知识令我终身受益。感谢侯勇、李矿伟、乔心旺三位师弟，谢谢你们给我的帮助，祝你们学业有成！

感谢我三年的同窗：高巍、刘冠良、王连玉、闫学斌、陈鹏飞。谢谢你们对我的帮助，祝你们家庭幸福、事业有成！

感谢所有关心帮助过我的人，谢谢你们！

个人简历、在学期间发表的学术论文与研究成果

个人简历

本人 1985 年 9 月 18 日出生于内蒙古赤峰市。2007 年 6 月毕业于南开大学信息技术与科学学院自动化系，获得工学学士学位，2007 年 9 月至今于南开大学信息技术与科学学院控制理论与控制工程专业攻读硕士学位。

研究成果

在研究生期间，本人参与了实验室横向课题《板卡式网络视频监控系统》的研制开发。本系统实现了视频实时预览、录像、回放、报警、移动侦测、遮挡、电子地图、云台控制等多种功能。我们运用面向对象程序设计、设计模式、软件测试及维护等多种技术，使这套系统实现了最初的设计要求，系统的稳定性和可扩展性都很好。我们开发的监控系统解决方案已经投入商业运营，如学校（浙江某医学院）、交通系统（某城市市内交通监控）、银行（天津市某行多个区属分行）、矿山（山西某煤矿安全监控）小区（某城市开发区数字城管监控系统）等领域。