

摘要

(随着多媒体技术及通讯技术的发展，视频监控系统正开始逐步在电气化铁路变电所采用，而当今最新流行的 Windows DNA 技术也已经开始在各方面得到应用，将两者有机地结合起来，实现最新技术的合并，必能获取最佳的收益。

本论文正是基于这种需求，从电气化变电所远程视频监控系统的调度监控中心软件开发入手，结合 Windows DNA 技术，分析了视频监控系统，开发了用于视频监控系统中用户管理、报警记录和操作记录管理的 COM 组件，设计了分布式视频监控系统的整体框架，对视频监控系统的安全性做出了详细的分析。)

本论文中探讨了 Windows DNA 在视频监控系统中的应用，包括如何设计三层结构中的数据服务层、商务逻辑层和展示层，并着重介绍了各层之间的安全性设置。其中数据服务层采用了 SQL Server，数据库访问采用了 ADO 技术；商务逻辑层采用了 COM 组件技术，而展示层则使用了 Windows 窗体和 ASP 网页。视频报警系统中的用户管理和报警记录、操作记录的管理是十分重要的，如何管理这些信息是 Windows DNA 三层结构都必须要考虑的问题。

本论文对 Windows DNA、COM 组件以及数据库管理技术进行了详细的应用研究，并实现了其中的三层设计，为 Windows DNA 技术在分布式视频监控系统的广泛应用进行了有益的探讨。

关键词：Windows DNA；COM 组件；视频监控系统；数据库

Abstract

With the development of the multimedia and communication technology, Video Monitoring System has been applied in the electrification railway transformer substation. And the most popular technology -- Windows DNA has been applied in many aspects of industry. It's no doubt that the cooperation between them would bring us more reward.

The thesis is written to meet such demand. It starts with Video Monitor Center Software of electrification transformer substation, analyses Video Monitor System combining the technology of Windows DNA, designs the COM component of user management, alarm record and operation record management used in Video Monitor System, analyses the whole framework of Distributed Video Monitor System, and makes a through analysis of its security.

The thesis discusses the application of Windows DNA in Video Monitor System and introduces how to design the three-tier architecture of the data service layer, business logical layer and presentation layer, and emphatically introduces security setting between these layers. SQL Server is implemented in data service layer and database access used by ADO technology. COM component technology is utilized in business logical layer, and Windows Forms and ASP Web pages are employed in presentation layer. User management, alarm records and operation records are very important in Video Monitor System. How to manage them in the Windows DNA architecture must be considered.

The thesis expounds the application of the Windows DNA, COM Component and ~~Video Management~~, achieve most functions of them. And it discusses helpfully the application of Windows DNA technology in the Distributed Video Monitor System.

Key Words: Windows DNA;COM Component;Video Monitor System; Database

第1章 绪论

1.1 理论及实际意义

COM 作为一种软件模型，不仅提供了程序与程序之间通信的标准，而且使我们改变了传统的程序设计方法。COM 结合了对象技术和组件技术两种特性，对象特性使得应用系统的设计和实现更加符合现实世界的面貌；组件特性使得应用系统可以充分发挥组件的优势，以适应现在应用的需要。

从软件业的发展历程来看，程序设计方法经历了多次变革^[39]，从最初的功能分解法，到结构化程序设计方法，再到至今仍在广泛使用的面向对象程序设计方法。人们对程序设计方法的选择取决于程序设计方法对应用软件的适用性。早期的应用软件以计算为主，所以功能分解法可以满足应用的需要；随着应用软件复杂程度的不断提高，结构化程序设计方法在很长时间内一直是软件设计人员的重要工具；面向对象程序设计是一个概念上的飞跃，它使得程序设计更加符合应用的本质结构，并且对软件产品的整个生存周期都起到了指导作用。

1.1.1 组件的概念和特点

组件^[2,3,7]不是一个新概念，在许多相对成熟的工程领域中，组件的概念是十分广泛的。在一个理想化的组件系统中，用户通常并不了解组件内部的知识，然而他们却可以通过配置组件完成所需的工作。例如由多个组件组成的 Hi-Fi 系统，“发烧友”并不需要了解扩音器、音箱、CD 播放器的制造原理，而只需要了解如何将其组装在一起，如何控制音量、均衡等。

软件行业的组件系统比其他的行业发展得要慢，不过，目前基于组件的程序设计方法正受到越来越多的重视和研究，像 COM 这样的组件体系也渐趋成熟。这些组件体系将形成新的软件开发平台，组件化的程序设计思想将使软件产业带来新的生机。

组件的两个重要特性：

(1) 组件之间可以在二进制级别上进行集成和重用。在一个成熟的组件应用系统中，组件往往是来自不同的软件独立开发商。在这种情况下，获得全部或部分源代码几乎是不可能的，同时也是不必要的。

(2) 组件必须被独立地生产、获得和配置。与环境和上下文无关是组件封装性的体现。

组件概念中对独立性和二进制形式构造方面的要求，使得诸如类型定义、宏、C++模板等等软件抽象形式都不能满足组件的基本要求，但是过程、类、模块、甚至是整个应用程序都可能满足组件的定义，只要它们满足组件的基本要求。

组件接口描述了组件对外提供的服务，组件和组件之间、组件和客户之间同时通过接口进行交互。因此组件一旦发布，它只能通过预定义的接口来提供合理的、一致的服务。这种接口定义的稳定性使客户应用开发者能构造出坚固的应用。一个组件可以实现多个组件接口，同时一个特定的组件接口也可以被多个组件来实现。

组件接口必须是能够自我描述的。这意味着组件接口应该不依赖于具体实现，将实现与接口分离彻底消除了接口客户和接口实现者之间的耦合关系，增强了信息的封装程度。同时这也要求组件接口必须使用一种与组件实现无关的语言进行描述。目前组件接口的描述标准是 IDL 语言。

由于接口是组件之间的协议，因此，组件的接口一旦被发布，组件软件厂商就应该尽量地保持接口不变，任何对接口的语法或语义上的改变，都可能造成现有组件和客户的联系遭到破坏。

1.1.2 基于 COM 的组件化程序设计方法

组件化程序设计^[1,4,8]方法继承并发展了面向对象程序设计方法。它将对象技术应用于系统设计，对面向对象程序设计的实现过程作了进一步抽象。我们可以把组件化程序设计方法用作构造系统的体系结构层次的方法，并且可以使用面向对象的方法很方便地实现组件。

组件化程序设计方法强调真正的软件重用和高度的互操作性。它侧重于组件的产生和装配，这两方面一起构成了组件程序设计的核心。组件的产生过程不仅仅是应用系统的需求，组件市场本身也推动了组件的发展，促进了软件厂商的交流与合作；组件的装配使得软件产品有可能用类似于搭积木的方法快速地建立起来，不仅可以缩短软件产品的开发周期，同时也提高了系统的稳定性和可靠性，因为组件可以被单独测试。

可重用组件不同于对象库，组件库保存的是一些经过测试的组件，这些组件遵从标准或规范，并且组件库也包括这些组件的细节功能说明文档。从组件的生产和消费的角度来看，基于可重用组件库的软件开发模式一般包括六个彼此相互联系的过程，如图 1-1 所示。

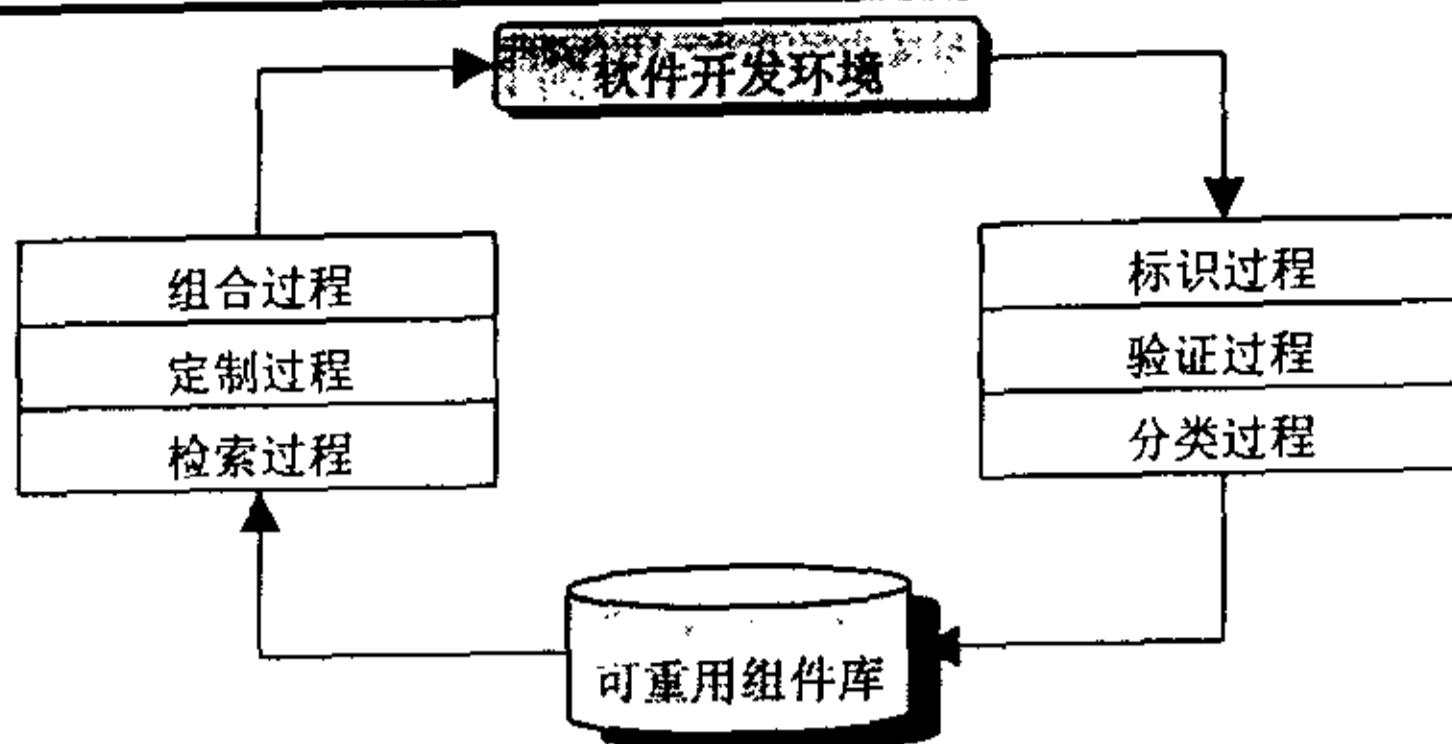


图 1-1 基于可重用组件库的软件开发模式示意图

下面对这六个过程作简单说明：

- (1) 标识过程：表示一个可重用的组件。
- (2) 验证过程：判定可重用组件所声明的功能特性和性能特性。
- (3) 分类过程：对已有组件根据它们的功能、使用方法、使用范围、接口等进行说明性的描述，以便日后可以高效准确的检索。
- (4) 检索过程：利用标准的方法根据指定的属性找到所需的组件。
- (5) 定制过程：通过对检索到的组件进行修改、裁剪和配置使其满足用户的需求。
- (6) 组合过程：将已定制好的组件之间集成在一起形成应用系统。

组件化程序设计方法在实施过程中必须遵守一定的标准或者规范，从目前的软件发展情况来看，比较成熟的规范为 COM 或者 CORBA。COM 和 CORBA 是两种相互竞争的标准，它们各自在一定的领域中取得了极大的成功，都有众多的软件支持这些标准。

我们以 COM 为基础，从组件的产生和构造角度出发，概括出组件程序设计方法有以下几个方面的特点：

(1) 编程语言和开发环境的独立性。在理想的组件软件体系中，用来构造应用系统的组件可能来自不同的软件开发商 (ISV, Independent software vendor)，而不同的 ISV 可能会选择不同的编程语言，这往往意味着调用方式、内存管理等方面差异；即使是同样的编程语言，不同的开发环境（如编译器）也会给组件的集成带来不便。COM 规范为组件建立了二进制一级的标准，不同软件开发商只需遵守 COM 规范，选择合适的编程语言。

(2) 组件位置的透明性。位置透明性是组件独立性的一个方面，对于分布式应用系统，位置透明特性可以跨越机器，也就是说，一台机器上的应用程序可以透明地访问其他机器上的组件。从 COM 到 DCOM 保证了组件的位置透明特性。

(3) 组件的进程透明性。进程模型是组件的基本特征，进程模型的不同选择会影响软件的运行效率。COM 通过列集机制很好地解决了进程透明性问题。

(4) 可扩充性。每个组件都是自主的，有其独特的功能，只能通过接口与外界通信。当一个组件需要提供新的服务时，可通过增加新的接口来完成，不会影响原接口已存在的用户。而新的用户可以重新选择新的接口来获得服务。

(5) 可扩展性(或可重用性)。COM 提供了两种重用模型：包容和聚合，在实现组件过程中，我们可以利用这两种方法对组件库中的组件进行扩展以得到新的组件。

(6) 具有强有力的基础设施。为了使组件有机地胶合在一起，实现无缝连接，需要功能很强的基础设施。这些基础设施是获得可重用性、可移植性和互操作性的有效工具。基础设施提供了诸如检索组件、启动远程组件等服务。

(7) 系统一级的公共服务。一个组件往往只提供相对较小的服务，而且在提供服务的过程中，可能会依赖于特定的、常用的公共服务，这种依赖性势必降低组件的封装特性。如果每个组件都必须实现这样的服务，则很可能阻碍组件之间的互操作性。但是如果这种服务在任何情况下都是可以访问的，那么这种依赖性就是可以接受的。COM 提供了诸如命名(Naming)服务、永久对象(Persistent)服务、统一数据传输等公共服务。

1.2 课题来源及主要研究内容

如今，生产过程中的无人管理已经成为当今电力行业的发展趋势，过去人们曾为实现对现场的设备进行控制及调节做出努力，这就是通常所说的“四遥”——遥测、遥信、遥控和遥调。

但随着多媒体技术及通讯技术的发展，已经很成熟地实现了将图像、声音信号的数字化，以及对音像信号的处理和远距离传送。显然这一技术可以使生产过程无人化中除“四遥”外，另增一项“遥视”功能。当然，遥视不仅是指在调度中心能够观看到生产现场的实景，而且还需要具备启动安全设施的功能，如启动消防系统、排水系统以及音响警告非法闯入者等。

视频监控系统已经开始在各种电气化接触网变电所使用，而 Windows DNA 技术也已经开始在各个方面得到应用。如何将两者有机地结合起来，以实现最新技术的合并，达到最佳的收益。

本论文就是以电气化接触网变电所远程视频监控系统开发为基础，对调度监控指挥中心如何实现视频数据和控制命令的发送、用户信息的管理以及报警记录和操作记录的管理等，对分布式视频监控等技术进行研究与开发实现。

本论文详细分析了 DNA 与视频监控结合的好处，以及实现视频监控系统的三层架构，并对各层的功能和实现做出了详细的设计，完成了 Windows DNA 技术在

视频监控系统中的应用，初步达到了分布式视频监控功能。

第 2 章 Windows DNA 架构

2.1 Windows DNA 定义

Microsoft Windows 的 Distributed interNet Applications (DNA: 分布式网间应用程序) 体系结构^[48]代表了 Microsoft 用以创建高分布式应用程序的方法。DNA 应用程序采用的是一种逻辑上三层的、基于组件的体系结构。Microsoft 的系统服务提供了这种类型的应用程序所需要的基础结构。而关键的基础结构服务则是 Component Object Model (COM: 组件对象模型) 和 Microsoft Transaction Server (MTS: Microsoft 事务服务器)。组件对象模型提供了组件交互作用的基本机制，而 Microsoft 事务服务器则为构造不同规模的服务器应用程序提供了工作环境。Windows DNA 作为一种综合系统结构，描述了如何为 Windows 平台创建三层应用程序。

Microsoft 推出 Windows DNA 这一套体系结构是为了创建一个基于 Microsoft Windows 平台的应用程序的框架，而这套结构则能够将个人计算机和 Internet 统一和集成起来。Windows DNA 打开了一个能够同时充分发挥个人计算机和 Internet 的能力的新模式。

在最高的层次上，Windows DNA 通过将核心服务高度集成到操作系统中，使计算机在内部网和公共网上都能够充分发挥协同工作的能力。这使我们的开发人员可以更容易地创建复杂的，能够支持大量用户的网络应用程序。更加重要的是，Windows DNA 提供了一种基于开放式协议和公共接口的具有高度协作能力的框架，并允许客户对现存的系统扩充新的功能，如 Web。这种开放的模型具有很强的扩展功能，第三方的厂家也可以创建他们的兼容的产品以扩展的这一结构。

Windows DNA 应用使用一套标准的基于 Windows 的服务，可以满足各个层次最新的分布式应用程序的需求，如用户界面和导航，业务处理和存储等。

Windows DNA 的核心是将 Web 和客户机/服务器的应用开发模型通过一套公共对象模型集成起来。Windows DNA 使用一套公共服务集，如组件、动态 HTML、Web 浏览器和服务器、脚本、事务处理、消息队列、安全性、目录、数据库和数据访问、系统管理以及用户界面等。这些服务在各个层次上以同一种方式开放给应用程序使用。

另外由于 Windows DNA 是在 W3C 和 IETF 等制定的标准上构建的，所以它充分支持开放式的 Web 功能。

利用 Windows DNA 提供的优势，开发人员可以构建全新的应用程序，包括电子

商务和其它个人之间及企业之间的通信应用，由于充分利用了标准网络服务和先进的基于组件的开发模式，开发人员可以比以往更快地开发出全新的应用。Windows DNA 支持和旧有的计算机软硬件之间的集成和扩展，所以 UNIX 和一些大型机的系统在 Windows DNA 模式下仍然可以使用。

Windows DNA 为网络应用提供了所有最基本的组件，所以开发人员可以把精力集中在产品的创新上，而无须关注底层结构和服务的构造。今天 Windows DNA 大部分的组件已被 Microsoft 的客户和合作伙伴广泛地利用了。随着 Microsoft Windows 2000 操作系统和 COM+（组件对象模型 COM 的增强版）的出现，Windows DNA 的结构和组件更易于被开发人员创建和使用。

Windows DNA 的特征如下：

1. 网络实现 Windows DNA 应用程序能利用 Internet 的一些关键的新应用，如全球信息网(WWW)。通过微软 Internet 信息服务器，Windows DNA 应用程序能提供用动态 HTML 语言创建的用户界面(HTML 为创建交互式万维网界面的标准语言)。这就提供了一个简化的浏览器界面、中央维护系统和让用户随时使用最新软件的新软件及时分销模型。针对微软 IE5.0，Windows DNA 应用程序支持多种用户操作系统，包括 Windows 2000，Windows NT，Windows 98/95，Macintosh 和 UNIX。

2. 管理 通过微软公司的零管理，就能对 Windows DNA 结构和构件元素进行集中管理，这就避免了网络管理员在安装和更新应用程序时对每个用户的 PC 机进行手工调整。

3. 动态化 通过 COM+的新的拦截增强机制，Windows DNA 结构能动态地改变其配置和功能。因此当企业条件和环境因素改变时，应用程序逻辑也可以自动改变。

4. 连接和移动性 当今 40% 的售于企业的 PC 机是膝上型电脑，它们在单机环境下使用的。对它们来说，最重要的是对离线操作的支持。Windows DNA 能使计算机连于互联网时将数据和功能智能地分割和存储，因此用户可以在脱离网络时利用企业的应用程序。

5. 网内和网外透明 因为 Windows DNA 把 Internet 和客户/服务器计算系统模型集成在了一起，用户可以象浏览 Web 主页一样地浏览应用程序。通过为联接计算提供单个、集成结构(联接计算即连接协作局域网(LAN)(Intranet)和 Internet 的计算系统)，Windows DNA 降低了维护和培训费用以及 PC 主人的其他一切成本。

6. 集成和发展演变 Windows DNA 结构不仅与网络相连，而且与协作数据和现有应用程序相连。通过使各组织共享数据和应用逻辑，并且在现有系统上建立界面以便于利用 Web 进行系统规划，Windows DNA 就可以在现有系统上增加投资，而不是另外建立单独的数据和应用的基本结构。

总之，Windows DNA 集成了各种微软客户必须支持的计算机系统投资，包括一系列用户界面模型：从基于主页的 HTML 应用程序到基于代码的 Windows 应用程序、

移动性用户的和脱机用户、结构式的和非结构式的存储、基于 Web 的和基于客户/服务器的应用程序。此外，DNA 在提供中央管理设施来降低布局和支持的成本时使用了分布式计算资源。

2.2 Windows DNA 的体系结构

开发客户机/服务器或者浏览器/服务器模式的应用程序，可以采用 Windows DNA 三层方案。Windows DNA 是 Windows 平台的应用程序开发模式。基本上，Windows DNA 提供了综合和集成的整套服务，所以开发者就可以从为分布式应用程序的建立或者组成所需要的结构负担中解脱出来，而把精力集中在商务解决方案上。

Microsoft 倾向于采用 Windows DNA 体系结构来编写分布式应用程序^[40~48]。这种应用程序可以在任何网络上发布。Windows DNA 的目标是使 PC、客户机/服务器和基于 Web 的应用程序围绕这一公共的应用程序体系结构合为一体。这种目标对任何大小的应用程序都是适宜的。

Windows DNA 三层架构包含了展示层、商务逻辑层和数据服务层，如图 2-1 所示。

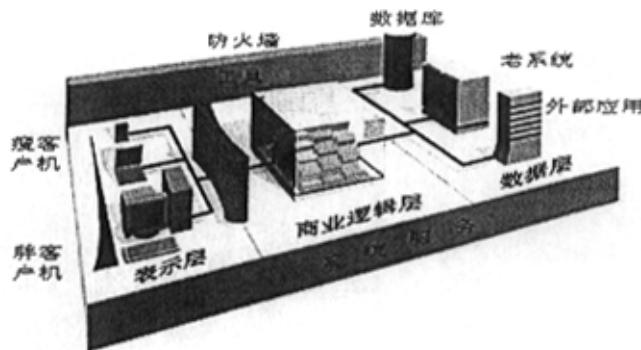


图 2-1 Windows DNA 三层结构示意图

1. 展示层

客户机/服务器的客户机端通常要完成用户接口和驱动系统的商务逻辑功能，只留下数据库给服务器端。这种设计可能导致加重捆绑在特定操作系统上的客户机应用程序的负荷，并且很难部署和支持。在三层架构中，客户程序设计为尽可能地“瘦”，一般只处理用户接口。这样的瘦客户机可能包括用 Visual Basic 设计的表单或者只是为运行在诸如 IE 之类的浏览器上而设计的 HTML 页面。

开发仅由 HTML 页面组成的客户机端应用程序对多数企业是很有吸引力的，这是因为它们的平台独立性并且分布灵活。开发比只有单纯的 HTML 页更加丰富的用户

接口的应用程序，可能考虑使用动态 HTML 或者脚本代码（IE 支持 VBScript 和 Jscript）或者在其 Web 页中包括 ActiveX 控件或 Java 小程序。与 HTML 一样，Java 小程序也与平台无关，并且，Java 小程序和 ActiveX 控件都提供自动分布功能。需要更稳定展示层的应用程序可以包括 Visual Basic、Java 和 C++ 等在内的功能完备的编程语言来创建。虽然用 HTML 创建的客户机程序可以使用超文本传输协议（HTTP）与 Web 服务器通信，用 Visual Basic、Java 和 C++ 等创建的应用程序则可以通过直接的方法调用运行在 COM/DCOM/COM+ 环境内的商务组件（Business Component）。

2. 商务逻辑层

虽然客户机/服务器架构适合于在不同计算机上部署客户机端和服务器端组件，但是三层设计的商务逻辑组件也可以产生更加灵活的解决方案。例如，应用程序的商务逻辑可能作为进程间 COM+ 组件而得到实现。这类组件设计成运行在客户端的客户机应用程序的进程上，或者在服务器端的 Web 服务器进程之上。同样，商务逻辑组件也可以运行在第三方机器的 COM+ 环境内，该计算机与客户机和数据库服务器都是分开的。

3. 数据层

Windows DNA 的数据层由 SQL 服务器组成，例如 SQL Server、Oracle、Sybase、DB2 或者其他支持 OLE DB 或开放数据库连接（ODBC）。通常，运行在中间层的 COM/DCOM/COM+ 组件使用 ActiveX 数据对象（ADO，提供 OLE DB 高级封装的 COM/DCOM/COM+ 组件）来连接并查询数据库。OLE DB 使得访问广泛的数据服务器（包括继承下来的旧有系统）上的数据成为可能。

2.3 视频监控系统中 DNA 构架的构建

2.3.1 视频监控系统的物理构成

视频监控系统^[10,14,16]由三大部分构成，包括监控远端（变电站）、通信网络和监控中心，如图 2-2 所示。图像传输介质可以用专线、电话网，也可使用微波、扩频或者光纤等通道，不同传输介质决定了传输图像的速度和质量。在监控远端和监控中心除了必要的摄像机、云台、矩阵外，大部分功能都有多媒体电脑完成，它可以方便地对图像进行任意处理、加工和传送，所以系统功能的扩展和改进可以完全通过软件进行。

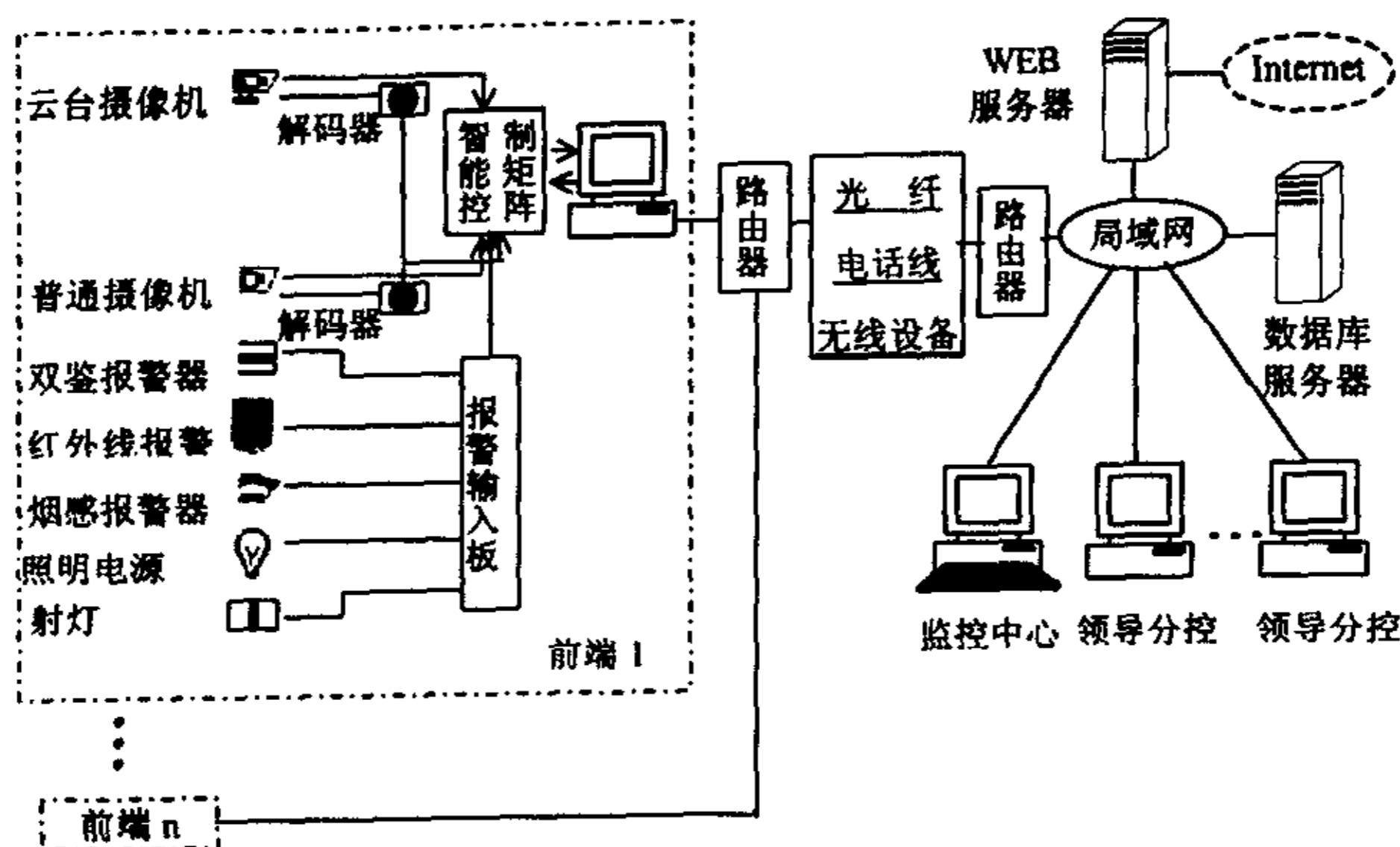


图 2-2 视频监控报警系统框图

监控远端的主要作用是：(1) 负责将摄像机采集的图像信号数字化，并将图像数据进行压缩，供录像和远程传输使用。(2) 远端还负责接受监控中心的远程控制命令，用以对现场的监控设备或者其它仪器进行远程控制。(3) 当该远端任何一个监控区域报警探头检测到警情时，主动切换到该报警区域，并将现场图像录制到本地硬盘上，并立即与报警中心建立通信联系，进行远程图像报警。(4) 负责相关数据的测量与传输如环境温度、触点温度等，并在达到危险值时发出报警。

远端的设备主要由监控区域设备、智能控制矩阵、多媒体计算机及多种附加设备等构成。监控区域设备主要包括各种类型的摄像机、解码器、各种类型的报警探头、以及各种可由本系统控制的现场设备。智能控制主机是监控区域设备和前端监控多媒体计算机的连接枢纽。所有监控区域摄像机的视频信号、解码器的控制信号、报警探头的报警信号以及设备开关的控制信号均接至智能控制主机。远端多媒体计算机是一台具有图形加速和视频图像采集功能的多媒体计算机，根据情况的不同，监控远端可能需要一些附加设备。如果监控中心希望同时观看某个变电所的多个监控区域，就需要在远端安装一台多画面分割器（常用的有四画面、九画面等分割器）。如果需要进行双向语音通话，则应增加一套语音通话设备，包括分布于不同区域的麦克风、喇叭、音频切换器、扩音器和语音通信设备等。红外测温功能则可以通过增加红外测温模块，红外测温仪器可以安装于预置云台上，事先设置好测温点，可以定期自动循环检测，也可以手动任意位置测量。多路测温需要增加多路测温数据读取控制设备，测温结果也可以保存到数据库中；而且，也可以根据需要编制各种报表式曲线，高温或者温度过高可以发出报警。

通信网络^[11,12,13]则可以采用双绞线、电话专线、DDN、ISDN、光纤、微波、无

线扩频或者卫星线路等多种方式，在变电所和监控中心之间建立数据传输通道。

监控中心包括通信中心、接警中心和各办公电脑上安装的远程分控。其中，通信中心相当于电信局的程控交换机，用于处理接警中心和多个远程分控，同时对某一个变电所远端进行访问而造成性能下降和竞争控制问题；并可以支持多个分控与一个远端之间的任意交叉访问，即实现“一对多、多对多”的图像数据通信。接警中心可以由一台配置较好的多媒体计算机担当，接警中心采用电子地图界面，当报警发生时能够使用语音提示具体发生报警的位置，并且能够同时观看来自不同变电所的全动态彩色画面。接警中心除了能够接受报警和进行报警处理、多路同时报警等功能外，还具备对监控远端进行远程控制功能如云台转动、照明开关等。远程分控是在主管领导或者生产部门的办公室电脑里安装远程分控软件来实现，并且经授权还可控制远端的设备，如镜头、云台、照明或者空调等。

从物理上讲，通信中心、远程分控都是连在专业网上的普通工作站。其区别是其网上的 IP 地址不同。从数据流上看，三者具有如图 2-3 所示关系：

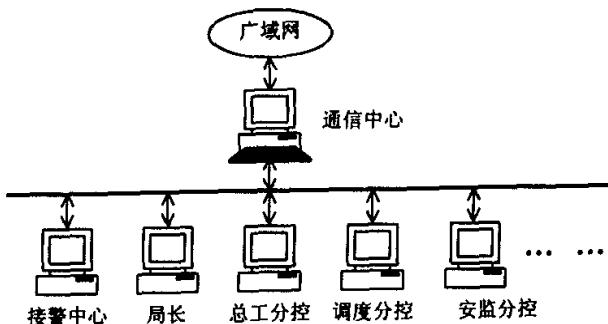


图 2-3 监控中心关系图

即所有数据都要通过“通信中心”转发，这样做的目的是为了提高多个分控同时访问一个远端的通信速度以及解决控制优先权的问题。

2.3.2 视频监控系统的 Windows DNA 设计

通过使用 Rational Rose 2000 设计出了视频监控系统的 Windows DNA 应用程序体系结构的概览图^[6,28,42,43,47]，如图 2-4 所示。

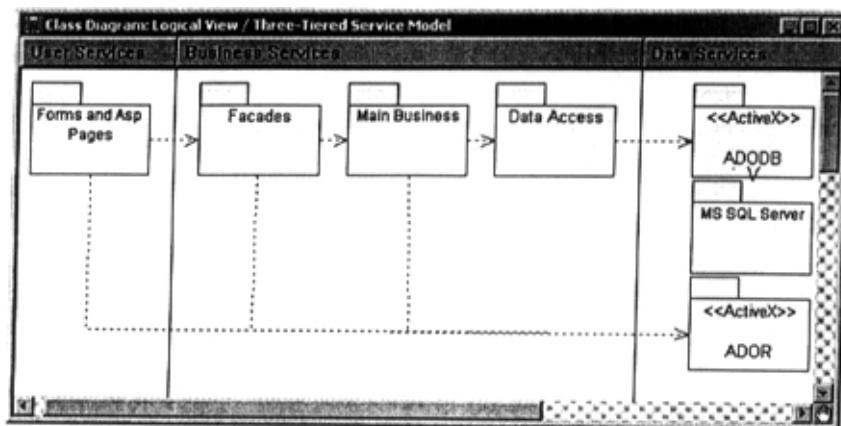


图 2-4 视频监控系统的 Windows DNA 概览图

该概览图使用 UML 语言来描述的。图中文件夹的符号称为包（Package），它们是 UML 包。UML 包是 UML 类模型中的一个逻辑容器，它能帮助区分模型中的类。

从图中我们将视频监控系统设计为三层：展示层（也称为用户服务层）、商务逻辑层和数据服务层。

1. 展示层 展示层只有一个包。依照这个模型，这个包中包含有不同类型的用户服务类。其中一部分是普通的 Windows 窗体（Forms）、动态 HTML 或 ASP 网页，也可以是 Visual Basic 中新的网页类（Webclass）的实例。

视频监控系统监控前端和监控中心中的前端界面都是 Windows 表单（Forms），而 Web 页面则是一般的 HTML 网页和 ASP 网页。这些客户机端应用程序只要通过直接调用商务逻辑层中设计好的 COM+组件就可以了。

2. 商务逻辑层 商务逻辑层包含有三个包：

(1) 外观包（The Facades Package） 这个包是用户界面需要与之通信的“真实”业务对象前面的外观类。外观对象是体系结构中唯一能由窗体或 ASP 网页访问并与之通信的元素。应用程序的其他部分对窗体不可见，包括主营业务对象、数据访问对象和数据库。

视频监控系统中外观类主要是用户管理组件和视频记录管理组件所提供的属性和方法。这些属性和方法在展示层中都可以直接查看调用，从而控制底层数据库。

(2) 主业务包（The Main Business Package） 该包包含了实体对象和面向集合的对象。这些对象为外观类提供服务，并依赖于提供数据访问操作的数据访问包中的对象。它们也依赖于提供数据传输服务和提供可能的状态保持服务的 ADOR 包中的对象。

在视频监控系统中实体类主要有报警记录类、操作记录类等。当获取一个类信

息（如报警记录）后，必须将其先保存到数据库中，否则当有一个类信息到来时，原信息就会丢失。同样，在两个组件（用户管理组件和视频记录管理组件）中，都有记录数据库信息和服务器信息的类。在两个组件中，实现了多种业务操作，包括创建新用户、删除用户、返回指定报警记录等。特别的是，在返回的类型为记录集时都采用了断开的 ADOR Recordset 对象。

(3) 数据访问包 (The Data Access Package) 该包建立 SQL 语句并进行存储过程调用，它包括从数据库中取数据的对象和其他修改数据库的对象。这些对象为主业务包中的对象提供服务，并且依赖于整个 ADODB 包，因为它们必须能够访问数据库。这个包同样管理业务。除了数据访问类，通常不允许其他类支持或请求事务，在其他包中的类应该远离这项任务。

3. 数据服务层 该层包括了三个包：

(1) ADODB 包 该包包含了所有在 ADO 中的对象并能完全访问 OLE DB，由此可以完全访问数据库。视频监控系统中使用了 ADO 来访问数据库，对数据库进行操作。

(2) ADOR 包 轻量级的 ADODB 并仅包含 ADO 记录集。只有数据访问包中的对象需要 ADO 的完全服务。所有其他对象用轻量级的 ADOR 能工作得很好，并且这样可以节省资源。

(3) MS SQL Server 包 由于在视频监控系统中使用的是 Microsoft SQL Server 2000。所以该包中代表了数据库本身。

视频监控系统采用的数据库服务器是 Microsoft SQL Server 2000，在组件中访问数据库时都是采用 ADODB，而返回的记录集则采用 ADOR，这样可以节省资源。由于 ADO 记录集需要在进程和机器之间移动，它使用调度方式来移动数据。移动数据需要遵循一定的规则：

1) ADO 记录集必须与数据库断开 通过将记录集的 ActiveConnection 属性设置为 Nothing 来实现。

2) ADO 记录集必须使用客户方的游标库 通过将记录集的 CursorLocation 属性设置为 adUseClient 来指定客户方游标。否则，不能把记录集和数据库断开。如果记录集保持连接，它肯定使用服务方游标，那么记录集拒绝在体系结构中的及其之间移动。

这些规则在视频监控系统中设计商务组件时都使用到了。

第 3 章 数据服务层的设计

3.1 视频监控数据库的设计

我们首先要进行数据建模。数据建模是建立用户数据视图的模型的过程，它是开发有效的数据库应用的最重要任务。如果数据模型不能正确地表示用户的数据视图，用户就会发现应用是难以使用的、不完整的和无效的。在数据库及其应用开发中，数据建模是随后工作的基础。

在数据库建模中，主要有两种不同的数据建模模型：语义对象模型（ORM 模型）和实体关系模型（E-R 模型）。

3.1.1 语义对象模型

语义对象模型是用来文档化用户需求并建立数据模型的。语义的含义是“有意义的”。语义对象的建模方法是用来部分地对用户数据的含义建模的。和关系模型比较，语义对象模型着重考虑用户对数据的理解，不是将精力主要花在提供一致的、高效的数据库存储和检索所依赖的物理结构的设计上，而是以进一步提高关系数据模型的层次为出发点，尽量使用户从数据库的物理细节中脱离出来。能从模拟真实世界实体或数据库环境的角度进行相对独立的操作，以便设计出较为实用的数据库结构。

语义对象模型中的语义对象是对用户工作环境中的某些可标识的事物的表示。语义对象的属性可以是单值的简单属性，可以是一组属性，还可以是一个对象。对象的每一个属性都具有最小基数和最大基数。最小基数是对象存在所必须的属性实例的数目，通常为 0 或 1；最大基数是指对象的属性实例的最大数目，通常为 1 或 N。

语义对象包括简单对象、组合对象、复合对象、混合对象、关联对象、父子类型对象和版本对象七类。这些对象的分类主要是根据对象的属性类型以及对象之间的关系进行的。简单对象是指只包含简单属性的语义对象；组合对象和复合对象分别指至少包含了一个属性组或一个对象属性的对象；混合对象至少有一个属性组和对象属性；关联对象是关联两个（或多个）对象并存储这种关联关系数据的对象；父子类型对象之间具有继承关系。各种不同类型的对象在数据库设计过程中的表达方式不同，但都有各自的描述模式。比如，复合对象的关系可以有 1 对 1、1 对多

或多对多三种情况，这三种情况的表示方法各不相同。1: 1 的关系只需将两个对象的关键字互置；1: N 的关系将前者的关键字置于后者中；对于多对多的关系，应建立交叉关系对象。

采用语义对象建模方法进行数据库设计的过程首先是根据用户常用的一些表格或文档资料，确定对象和其属性，再通过对象之间的相互关系分析进行对象归类，不同类型的对象模型具有不同的数据库设计方法和模式，在此基础上可进行数据库的设计。

3.1.2 实体关系模型

在需求分析阶段，通过对应用环境和要求进行详尽的调查分析，用多层次数据流图和数据字典描述了整个系统。设计 E-R 图的第一步，就是要根据系统的具体情况，在多层次的数据流图中选择一个适当层次的数据流图，让这组图中每一部分对应一个局部应用，我们即可以这一层次的数据流图为出发点，设计分 E-R 图。

一般而言，中层的数据流图能较好地反映系统中各局部应用的子系统组成，因此人们往往以中层数据流图作为设计分 E-R 图的依据。

每个局部应用都对应了一组数据流图，局部应用涉及的数据都已经收集在数据字典中了。现在就是要将这些数据从数据字典中抽取出来，参照数据流图，标定局部应用中的实体、实体的属性、标识实体的码，确定实体之间的联系及其类型（1:1、1:n、m:n）。

现实世界中一组具有某些共同特性和行为的对象就可以抽象为一个实体。对象和实体之间是“is member of”的关系。例如在视频监控环境中，可以把烟雾报警、红外线报警等对象抽象为报警实体。

对象类型的组成部分可以抽象为实体的属性。组成部分与对象类型之间是“is part of”的关系。例如报警时间、报警名称、报警地点、报警类型等可以抽象为报警实体的属性。

实际上实体与属性是相对而言的，很难有截然划分的界限。同一事物，在一种应用环境中作为“属性”，在另一种应用环境中就必须作为“实体”。一般说来，在给定的应用环境中：

- (1) 属性不能再具有需要描述的性质。即属性必须是不可分的数据项。
- (2) 属性不能与其他实体具有联系。联系只发生在实体之间。

3.2 视频监控数据库中视频信息的设计

对于视频监控系统来说，数据库中主要存储的是各种用户信息、用户权限信息、各种报警记录、各种操作记录以及其他各种数据^[10,14]。

对于用户信息的存储，采用了组管理方案^[51]，这样可以增强数据库用户的安全性和数据库服务器的安全性。而对于报警记录和操作记录的存储，只要将记录的详细信息保存起来就可以了，只是由于报警纪录涉及到报警录像文件的存储，因此需要特别考虑。在视频监控系统中，当发生报警时，除了系统自动将报警信息保存到数据库中外，系统也要自动将报警记录录像保存到硬盘上，所以数据库中只要保存报警录像文件的位置就可以了。

对视频数据库的设计，采用了 PowerDesigner 7 工具设计好数据库的 ER 图，然后生成 SQL 文件，以此来创建数据库，包括创建数据库中的各种对象（表、视图、存储过程等）。

首先建立视频监控数据库登录帐户。由于在视频监控系统中使用了四种用户类型：管理员、领导用户、一般用户和前端用户，这四种用户在 Windows NT 对应于全局域用户组，其中 Domain 为域名。在 Windows NT 域管理器中创建了四种用户组：视频监控系统管理员组（Admins）、视频监控系统领导组（Leaders）、视频监控系统一般用户组（ClientUsers）和视频监控系统前端用户组（ForeUsers），分别对应于视频监控系统中的四种用户类型。SQL Server 服务器采用了 Windows NT 认证模式，所以使用这四种用户都可以登录到服务器上。所以在 SQL Server 视频监控数据库中添加了这四种用户，如图 3-1 所示。

视频监控系统中设计了两个存储过程 sp_alarmrecord、sp_runningrecord，分别用来根据用户传送不同的参数值返回不同的报警记录或操作记录。其参数取值含义如下：

- 0 — 返回所有的报警/操作记录
- 1 — 打开最新的 20 条报警/操作记录
- 2 — 打开最新的 30 条报警/操作记录
- 3 — 打开最新的 50 条报警/操作记录
- 4 — 打开最新的 100 条报警/操作记录
- 5 — 打开最近 5 天内的报警/操作记录
- 6 — 打开最近一周内的报警/操作记录
- 7 — 打开最近半个月内的报警/操作记录
- 8 — 打开最近 30 天内的报警/操作记录

这样，就可以向这两个存储过程传送不同的参数值以返回我们所需要的记录信息。

视频监控数据库中各个对象的 ER 关系图如图 3-2 所示。

登录用户表 Login 用于保存视频监控系统中所有登录用户的信息，包括以下字段：记录 id，用户真实姓名，用户权限，用户名，口令，是否禁止登录，当前是否登录，组 id，备注。其中记录 id 为主关键字，组 id 为外关键字。

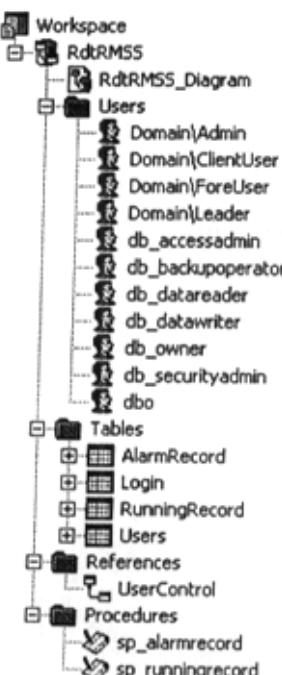


图 3-1 视频监控数据库对象信息

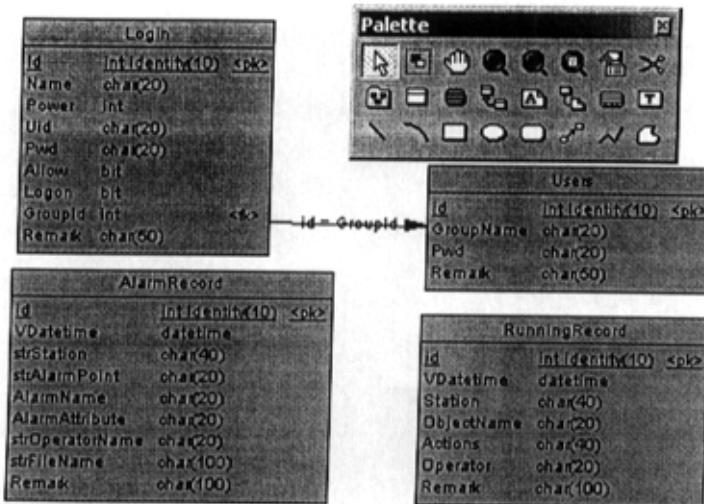


图 3-2 视频监控数据库的 ER 模型

组表 Users 用于存放登录 SQL Server 服务器的登录用户帐户信息，包括以下字

段：组 id，组名（即登录用户帐户名称），组口令，备注。其中组 id 为主关键字。

报警记录表 AlarmRecord 用于保存各种报警记录信息，包括以下字段：记录 id，报警时间，报警站点，报警点，报警名称，报警属性，操作员，报警录像文件名，备注。其中报警录像文件名用于保存当前报警的报警录像文件名位置，而备注用于注明当前报警的特殊信息，包括特殊的报警原因等。其中记录 id 为主关键字

操作记录表 RunningRecord 表用户保存用户的各种操作记录信息，包括以下字段：记录 id，操作发生时间，站点，操作目标对象，动作名称，操作员，备注。其中记录 id 为主关键字

当使用 PowerDesigner 设计好视频数据库的各种对象后，就可以使用其“Generate Database”命令，产生 SQL 文本（SQL Script），然后就可以在 SQL Server 服务器上运行该 SQL 文本，以产生设计好的数据库及各个数据库对象。或者在命令行中直接输入：

```
isql -Usa -P*** -Ssandy -iSqlScript.sql
```

其中，sa 为数据库服务器的超级管理员帐户，***为管理员帐户口令，sandy 为数据库服务器名称，SqlScript.sql 为由 PowerDesigner 生成的 SQL 文本文件。当执行该命令后，服务器就会自动创建视频监控数据库中的各个视频信息对象。

3.2.1 采用组管理

在 Windows NT 或者 Windows 2000 的用户管理中，都是采用将用户划分为不同的用户组，组中的所有成员都继承用户组的所有权限。一个用户可以同时属于多个组。这种组管理可以简化权限分配。因此在视频监控系统设计中，也采用组这种管理特性，并通过在数据库中建立唯一的角色与之对应。本设计中的组概念与一般的组概念稍有不同，这里所谓的组对应着数据库的登录用户帐户。当用户从数据库表中获取其对应的组后，然后再使用该组（即数据库登录用户帐户）连接数据库，这样保证了所有该组的成员登录到数据库后，它们的权限是相同的。

由于在视频监控系统中，主要涉及到四种用户，所以我们只建立者四种数据库用户帐户就可以了。

以建立管理员用户帐户为例：

```
sp_grantdbaccess 'Domain\Admin' , 'RdtRMSS'
```

建立视频监控数据库 RdtRMSS 的用户帐户可以为 Windows NT 用户帐户‘Domain\Admin’。其中‘Domain\Admin’用户属于 Windows NT 的视频监控系统管理员组‘Domain\Admins’中成员。

定义好用户帐户后，就需要对数据库对象进行管理许可设置。由于管理员对报

警记录表和操作记录表具有完全访问的权限（查看、增加、删除、更新等），所以对象许可设置如下：

GRNAD ALL ON AlarmRecord TO Domain\Admin

GRNAD ALL ON RunningRecord TO Domain\Admin

对于其他三种用户，对报警记录表和操作记录表只具有查看权限，许可设置如下：

GRNAD SELECT ON AlarmRecord TO Domain\ClientUser

其中‘Domain\ClientUser’是 Windows NT 中视频管理系统一般用户组中成员。

3.2.2 用户帐户管理及组管理

登录用户属于应用系统中的一个组，其与数据库中定义好的组关系可定义成数据库表，如表 3-1 所示。

表 3-1 登录用户与数据库内组对应表

登录名	口令	组名	说明
管理员	*****	Administrators	管理员
李强	*****	Leaders	领导组
王容福	*****	ClientUsers	一般用户组
赵松峰	*****	Foreusers	监控前端组
.....			

表 3-2 组与数据库用户帐户对应表

组名	数据库用户帐户	口令
Administrators	Domain\Admin	*****
Leaders	Domain\Leader	*****
Foreusers	Domain\Foreuser	*****
.....		

将整个视频监控系统划分为监控前端、监控中心等模块，按各个模块分成几个组，同时要根据模块内的操作权限不同再将这几个组划分的更细，如将管理中心组分成领导组、管理员组等。当用户一登录到系统中，就根据该表确定出其所属的组，同时也确定了该用户的权限。

定义好的组与数据库的用户帐户一一对应，如表 3-2 所示。

在数据库中，预先建立一个特殊登录用户 LoginUser，给该用户只具有对上述两个表查看的功能。在开始连接数据库时，先使用该用户进行连接数据库，先从第一个表中判断该用户是否合法，如果合法，则从表中取出该用户对应的组名，然后根据组名从表 3-2 中获取对应的数据库用户帐户及口令。再使用该数据库用户帐户连接数据库，既可完成系统登录。

3.2.3 ActiveX 数据对象 (ADO)

OLE DB 对象模型通过那些与 Automation 不兼容的 COM 接口展示它的功能。这意味着在很多程序设计语言和工具（包括 Visual Basic 和 Microsoft Visual J++）中，不能直接使用 OLE DB。这样 UDA 也定义应用程序级的编程接口 ADO(ActiveX Data Objects; ActiveX 数据对象)^[31]。所有 ADO 接口都是双重接口，所以它们可被任何 COM 敏感型的程序设计语言或者工具使用。专家推荐采用 ADO 方式访问 Windows DNA 应用程序的数据存储。

ADO 对象模型如图 3-3 所示。ADO 构造在 OLE DB 之上，所以可以看到它与 OLE DB 对象模型有许多相似的地方。与 DAO 和 RDO 不同，ADO 对象模型不是层次型的。除了 Error 和 Field 对象外，所有其他对象都可独立独立创建。这一功能使得在不同环境重复使用对象变得比较容易进行。它也意味着，往往可用几种不同的方法完成同一个特定的编程任务。

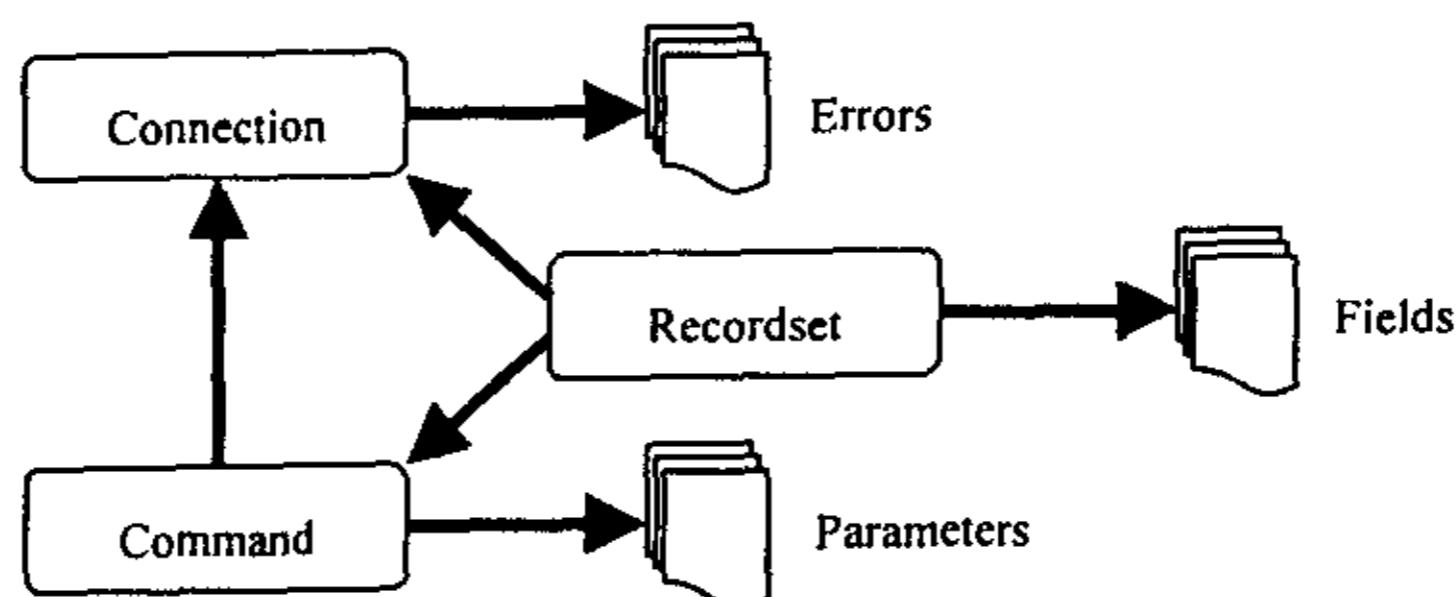


图 3-3 ADO 对象模型

Connection (连接) 对象代表到一个数据存储的唯一会话期 (Session)。Connection 实质上是 OLE DB 中的 Data Source 对象和 Session 对象的一种组合。Connection 对象表述一个 Execute (执行) 方法。利用这一方法，就可以花费最少的精力来执行简单的任务。另外，Connection 对象可以依附在 Command 对象和 Recordset 对象之上，它也提供访问数据存储中数据的方法。

ADO 中的 Command 对象等价于 OLE DB 的 Command 对象。它们两者都提供一种办法，用来针对数据源准备和执行参数化的命令。通过准备一个命令，您可保存该命令的一种处理后的格式，需要时可快速执行这一命令。一个 Command 对象有一个 Parameters(参数)集合，它包含一个或多个 Parameter 对象，每个这样的 Parameter 对象代表一个针对具体命令的参数。只有当基础 OLE DB 提供者实现了 OLE DB 的 Command 对象时，ADO 的 Command 对象才可用。

Recordset(记录集)对象是 ADO 的心脏。与 OLE DB 的 Rowset 对象类似，Recordset

对象代表的也是来自一个数据源的列表式数据。那些从数据存储中返回数据的 Connection 和 Command 方法，将返回只读的 Recordset 对象。这样的对象可被顺序地访问。程序员可直接创建更灵活地 Recordset。可将这样的 Recordset 连接到 Connection 对象，并且可以有选择地连接到 Command 对象。可调用各种 Recordset 方法来充实这样的 Recordsets。Recordsets 支持多个选项，它们用于控制在一段指定的时间内从一个数据源中检索出的数据量。还有一些选项可用来控制基础数据源的类型和对数据源实施加锁的持续时间，并指定何时更新数据存储。

Recordset 对象在逻辑上由若干组行和列组成。在任何指定的时间内，一个 Recordset 代表的是与一个特定的行（被称为当前行）相关联的一列。Recordset 中的各个具体的列通过其 Fields 集合访问，每列对应一个 Field 对象。对每个 Recordset，都有一个临时表与之相关联。用数据库的语言来说，临时表就是向应用程序返回数据行的软件。一个 Recordset 中的临时表指示当前在该 Recordset 中的位置，并确定将返回的下一行。ADO 支持几种临时表类型，包括简单的“单向向下（Forward-only）”临时表和允许移动到任何行的临时表，还有一些特别临时表，当在该 Recordset 中浏览时，可用它们查看其他用户对 Recordset 的修改。

3.3 数据服务层的安全性设计

数据服务层是应用程序后台的控制权限层。所有在应用程序前台的控制权限最后都要通过对数据库的权限控制^[52]来实现。在该层中，主要实现数据库的连接、数据库用户管理、角色管理、角色成员管理、角色授权等功能。所有这些功能只有管理员才具有。管理员通过创建数据库用户，并对该用户进行赋权。其他用户一般只有对几个表具有查看权限，至于一些敏感数据，只有管理员才能进行操作。

数据库服务器 SQL Server 的访问控制分为两个阶段：

- (1) 安全帐户认证
- (2) 访问许可确认

在安全帐户认证阶段，NT 操作系统或者 SQL Server 都可以对用户登录进行认证，如果认证成功，用户就可以连接到 SQL Server 上，否则服务器将拒绝用户的连接请求，如图 3-4 所示。

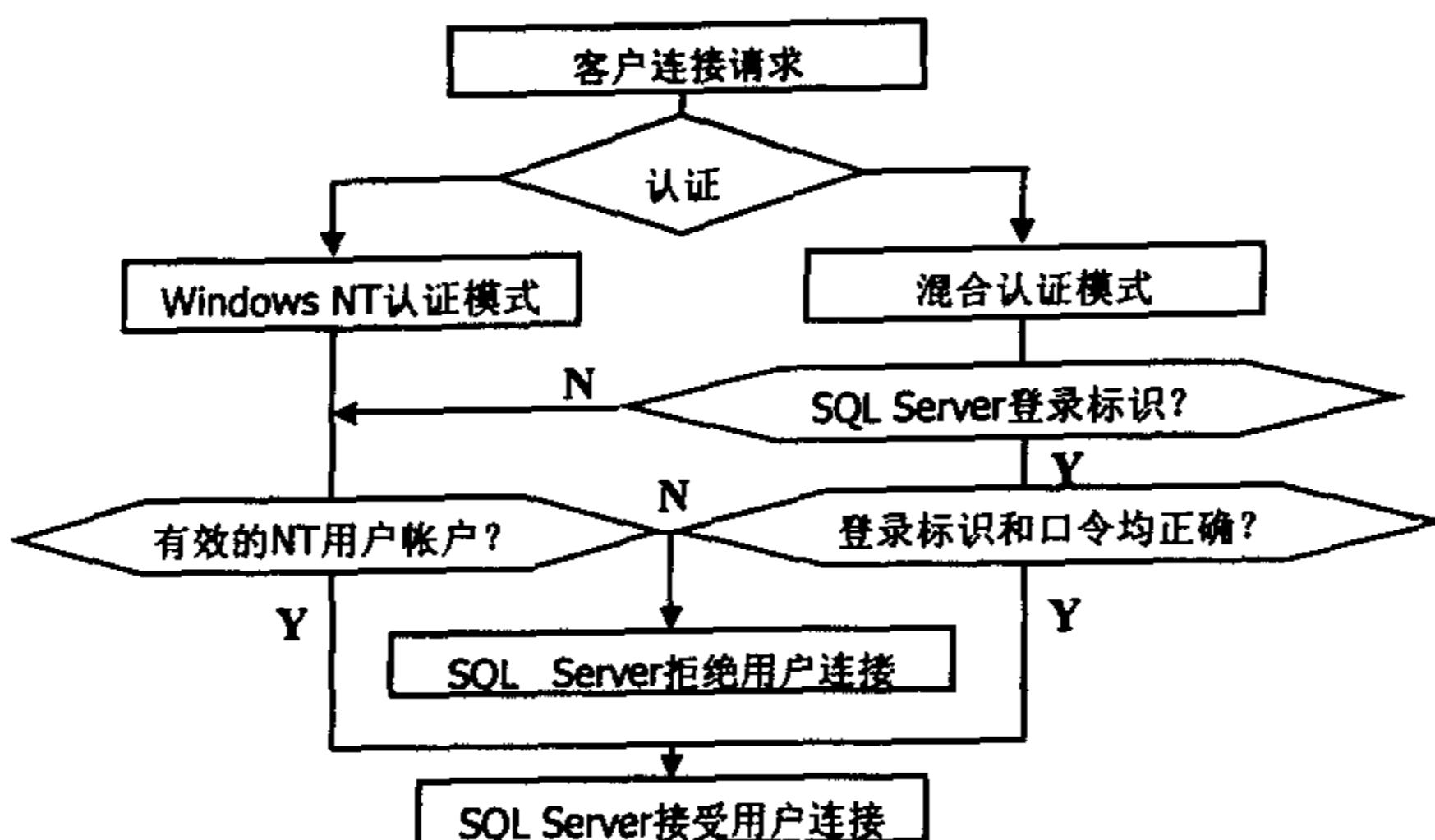


图3-4 SQL Server对登录标识的认证过程

用户成功连接到 SQL Server 后，系统根据数据库中保存的与服务器登录标识相对应的用户帐户来判断他们是否具有对数据库的访问许可。用户对数据库的访问许可被确认后才能访问指定的数据库对象，否则，SQL Server 将拒绝用户的访问请求。

在 SQL Server 中，服务器登录标识是由系统管理员创建的，而数据库对象的访问许可则由数据库对象的所有者授权。

在视频监控系统中，采用的是 Windows NT 认证，因为 NT 认证更为安全。Windows NT 认证具有安全确认、口令加密、审核、口令有效期保护、最短口令长度限制、非法登录时帐户锁定等功能。通过采用 NT 认证，就将 SQL Server 的安全性与 NT 的安全性统一起来。

视频数据库中登录用户表 Login 和组表 Users 的设计也保证了数据库中的各个对象的安全。这我们在商务服务层的组件设计过程中充分使用了这个技巧。

程序代码实例如下：

```

Set cn = CreateObject("ADODB.Connection")
Set rs = CreateObject("ADODB.Recordset")
'创建两个 ADODB 对象 Connection 对象和 Recordset 对象
ConnectionString = "Provider=SQLOLEDB.1;Integrated Security=SSPI; Persist Security
Info=False;User ID=LoginUser; Password=*****;Initial Catalog=RdtRMSS;Data
Source=ServerName"
cn.Open ConnectionString
'先以特殊登录用户 LoginUser 连接到数据库上
CommandString = "Select a.Name,a.Power,b.GroupName,b.Pwd From Login a,Users b Where
a.Uid=''' & Username & ''' And a.Pwd = '' & Password & '''"
  
```

```
rs.Open CommandString, cn, adOpenForwardOnly, adLockOptimistic, adCmdText
‘判断当前登录到视频监控系统的用户是否合法
If rs.EOF Then
    .....
    ‘如果该用户不合法，则返回错误信息，并退出函数
    Exit Function
Else
    Group = rs(“GroupName”)
    GroupPwd = rs(“Pwd”)
    ‘如果该用户合法，这获取其对应的组信息
    rs.Close: cn.Close
    ‘关闭当前的数据库连接
    ConnectionString = “Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;User ID=” & Group & “; Password=” & GroupPwd & “;Initial Catalog=RdtRMSS;Data
Source= ServerName”
    cn.Open ConnectionString
    ‘重新以获取的组用户（数据库登录用户帐户）连接到数据
    ‘库上，此时用户的权限就由该组用户的权限所限制
    CommandString = “*****”
    rs.Open CommandString, cn, adOpenForwardOnly, adLockOptimistic, adCmdText
    .....
    ‘重新生成 SQL 语句，获取用户所需记录信息，然后最获
    ‘取的信息进行处理
End If
```

第 4 章 商务逻辑层的设计

4.1 组件的基本概念及其安全机制

4.1.1 组件的特性

COM 规范所定义的组件模型，除了具有面向对象的特性和客户机/服务器特性这两个基本特性外，还包括如下特性：

1. 语言无关性 COM 规范的定义不依赖于特定的语言，因此编写组件对象所使用的语言与编写客户程序使用的语言可以不同，只要它们都能够生成符合 COM 规范的可执行代码即可。COM 标准与面向对象的编程语言不同，它所采用的是一种二进制代码级的标准，而不是源代码级的标准。在面向对象编程(OOP, Object-Oriented Programming) 语言中定义的对象，只能在同样的语言中被重复使用，这就大大限制了对象的重用。当然，OOP 语言可以被用于创建 COM 组件，因此这两种技术实际上 是互相补充的。COM 对象把 OOP 语言中的对象封装起来，并提供一致的接口，使得它可以被各种不同的语言所使用。例如，用 C++思想的 COM 组件中的对象，可以很容易地在其他的语言如 Java 中被使用。因此，COM 的语言无关性实际上为我们跨语言合作开发提供了统一标准。

当我们设计和开发一个应用系统时，在开始编写代码之前，首先我们要选择合适的语言，语言的选择对整个项目可能会产生很大的影响，而且，开发人员本身对语言也有喜好，让开发人员放弃他所熟悉的语言而去学习一门新的编程语言，并非明智之举。只要所选择的语言能支持 COM 组件的生成，这按 COM 组件模型构造的系统就允许使用此种编程语言。

虽然 COM 规范的定义与语言无关，而且差不多每种语言在实现时都提供了对 COM 的支持，如 Microsoft 的 Visual C++、Visual Basic、Visual J++，Borland 公司的 Delphi、C++ Builder 等都支持 COM 组件的开发和使用，而且很多语言还提供了许多可直接利用的 COM 组件作为产品的可选配件。但与 COM 规范最贴切并且最能够反映 COM 特性的还是 C 或者 C++语言^[32, 33, 34]。

2. 进程透明性 在客户机/服务器模型的软件结构中，运行在客户端的代码和运行在服务器端的代码，既可以在同一个进程中，也可以在不同的进程中。如果它们运行在同一个进程中，则由于组件和客户共享了进程的资源，因而无论对于编程还是运行效率都是很有益的。但实际情况往往并不是这样简单，因为服务程序并不总

是作为 DLL 被装入到客户进程中，它也经常是一个 EXE 可执行程序，因此，跨进程操作是很必要的。

COM 所提供的服务组件对象在实现时有两种进程模型：进程内对象和进程外对象。如果是进程内对象，则它在客户进程空间中运行；如果是进程外对象，则它运行在同一机器上的另一个进程空间或者在远程机器的进程空间中。我们通常也按下面的方式对组件对象服务程序进行区分：

(1) **进程内服务程序** 服务程序被加载到客户的进程空间，在 Windows 环境下，通常服务程序的代码以动态连接库 (DLL) 的形式实现。

(2) **远程服务程序** 服务程序运行在与客户不同的机器上，它既可以是一个 DLL 模块，也可以是一个 EXE 文件。如果远程服务程序是以 DLL 形式实现的话，则远程机器会创建一个代理进程。

虽然 COM 对象有不同的进程模型，但这种区别对于客户来说是透明的，因此客户程序在使用组件对象时可以不管这种区别的存在，只要遵照 COM 规范即可。然而，在实现 COM 对象时，还是应该慎重选择进程模型。进程内模型的优点是效率高，但组件不稳定会引起客户进程崩溃，因此组件可能危及客户；进程外模型的优点是稳定性好，组件进程不会危及客户程序，一个组件进程可以为多个客户进程提供服务，但进程外组件开销大，而且调用效率相对低一些。

实现这种进程透明性的关键在于 COM 库，COM 库负责组件程序的定位，管理组件对象的创建和对象与客户之间的通信。当客户创建组件对象时，COM 库负责装入组件模块或者启动组件进程，如果客户指定在远程机器上创建对象，则两台机器上的 COM 库会协作完成远程 COM 对象的创建工作，并且在客户进程中创建一个代理对象 (Proxy Object)，客户程序直接与代理对象进行交互。因此，客户程序可以不管组件对象的进程模型，即使组件的进程模型发生了变化，客户程序也不需要重新编译。

3. 可重用性 可重用性时任何对象模型的实现目标，尤其对于大型的软件系统，可重用性非常重要。而且，由于 COM 标准是建立在二进制代码级的，因此 COM 对象的可重用性与一般的面向对象语言如 C++ 中对象的重用过程不同。

对于 COM 对象的客户程序来说，它只是通过接口使用对象提供的服务，它并不知道对象内部的实现过程，因此，组件对象的重用性可建立在组件对象的行为方式上，而不是具体的实现上，这是建立重用的关键。

COM 用两种机制实现对象的重用。我们假定有两个 COM 对象，对象 1 希望能重用对象 2 的功能，我们把对象 1 称为外部对象，对象 2 称为内部对象。

(1) **包容方式** 对象 1 包含了对象 2，当对象 1 需要用到对象 2 的功能时，它可以简单地把实现交给对象 2 来完成，虽然对象 1 和对象 2 支持同样地接口，但对象 1 在实现接口时实际上调用了对象 2 地实现。

(2) 聚合方式 对象 1 只需简单地把对象 2 的接口递交给客户即可，对象 1 并没有实现对象 2 的接口，但它把对象 2 的接口也暴露给客户程序，而客户程序并不知道内部对象 2 的存在。

对象重用是 COM 规范很重要的一个方面，它保证 COM 可用于构造大型的软件系统，而且，它使复杂的系统简化为一些简单的对象模块，体现了面向对象的思想。

以上内容说明了 COM 规范的几个主要特性。COM 接口本身也是 COM 的一个特性，它为 COM 组件与客户之间进行通信提供了保障，通过引用计数（Reference Counting）的方法控制了组件对象的生存期，为多个客户共享同一个组件对象提供了有效的控制手段。

4.1.2 Windows NT 的安全机制

Windows NT 包含一个健全的安全子系统，能够处理验证、访问控制、检查和数据完整性。登录 Windows NT 时，就是与 NT 安全系统的组件 Winlogon 交互，然后这个过程将用户的用户名和口令转发给另一个子系统组件 Local Security Authority (LSA)。LSA 与几个其它子系统组件交互，负责验证用户名和口令正确与否。如果用户是一个域用户，则要通过域控制器进行验证。显然，这里用户名和口令是关键要素，必须保密。这里通过 NTLM (NT LAN Manager) 或者 Kerberos 之类的验证协议进行的，它不需要向域控制器发送实际口令。用户允许进入安全系统后，开始请求访问资源或功能。根据系统情况，有些用户可能和其它用户有不同的访问权限。

4.1.3 COM+的安全体系

COM+的安全性^[7, 24, 26, 31]是按照中间件、多用户以及企业规模组件系统的要求设计的，这是安全性考虑的顶级设计。

角色是整个 COM+ 安全构造中最核心的概念，一个角色代表一种用法。角色是在应用程序到应用程序这样一个基础上被定义的，角色可以由管理员根据企业的需要来定义，也可以由应用程序开发人员根据系统的需要定义，当然也可以根据二者的需要来定义。这样的话，程序开发人员可以在不知道大量的开发环境结构的情况下，确保程序的安全需求能够被很好地满足。为了使用他们的计算结构编制程序网，管理人员分配给系统用户和组以不同的角色，这些不同的角色又是为了某个应用程序而建立的。

角色由系统构造者合理地安排，并由配置组件的管理员配置后，两个系统之间

提供安全检查。NT 内建的安全性可以满足管理员配置安全性的要求。例如，管理员可以修改一个对于视频监控系统中所有一般用户都无效的类接口。组件能够提供它们自己的内部逻辑来控制访问。例如，视频监控系统中一般用户和领导可以允许除了删除报警记录方法和删除操作记录方法之外的所有例行程序调用，而这些方法只有管理员才可以调用。

程序设计中，一个简单的函数可以完成 COM+大部分的安全性检测工作：
IObjectContext::IsCallerInRole()。

COM+中保留着上下文数据，这些数据描述了每一个 COM+对象调用者的角色。通过检测调用者的角色，组件就可以确定调用者是否应该被允许执行该任务或者应该在多大程度上扩展其能力。COM+通过限制角色的权限简化了处理分布式对象环境时的复杂性，这对我们进行分布式应用程序设计提供了很大的方便。

COM+安全性包括两个基本的应用方法：声明安全性和过程安全性。

1. 声明安全性 声明安全性是 COM+安全性的一部分，可以通过管理的方法来定义。即管理员安排用户和组帐号到应用程序角色中。管理员也可以分配需要的角色到类和接口上，用于限定那些角色能够使用被保护的元素。

2. 过程安全性 过程安全性是通过组件代码，在编程过程中管理的安全性。一些应用程序可以毫不阻拦的通过声明安全性。重要的是，要注意到过程安全性使得程序开发人员可以在子方法级别上选择替代的逻辑性，而声明安全性仅仅用于接口级。

4.2 视频监控组件的设计

在组件分析设计过程中，我使用了 UML 技术^[9, 28]。使用 UML 技术，并不是说使用该技术分析和设计了整个视频监控系统（当然也可以），而是使用 UML 中的技术来描述和分析视频监控组件的功能，这样就可以清楚地了解到组件的工作流程和作用。

4.2.1 用户管理组件的分析

用户管理组件主要包括视频监控中各种用户的管理功能。在视频监控中，主要包括四种用户，监控前端用户、管理员、领导和一般用户。其中监控前端用户主要是处于监控地点最前端的管理用户，该用户对其监控的监控设备具有至高无上的权利，他可以查看他所监控区域的任何位置和任何设备，但对其他监控前端的监控设备则无任何控制权利，包括无查看权利。监控中心包括三种用户管理员、领导和一

般用户，其中管理员的权限最大，领导次之，一般用户最小。这三种用户都可以登录视频监控系统，查看监控情况。在多个用户同时设置监控属性时，以其中权限最高的用户设置为准。

由于监控前端用户的功能在前端机上，在前端机上运行的是单独的前端监控系统，其功能比较单一，对数据库的访问只有查看视频记录，而没有用户管理功能。所以用户管理组件中不包括对监控前端用户的管理。用户管理组件主要运用在视频监控中心。

视频监控中心的用户包括三种：管理员、领导和一般用户。管理员负责监控中心所有用户的管理，所以他必须能够创建用户、删除用户。又可以对用户进行授权和禁止，同时他又可以列举所有用户，以便可以对用户进行一些操作。

领导用户和一般用户在用户管理组件中没有什么特殊权限，只有在监控系统中，依据用户验证返回的权限来决定其操作限制。

当用户登录系统时，需要对用户进行验证；在用户退出系统时，也需要对当前用户进行注销；所以所有用户都有用户验证和注销功能。登录系统后，用户可以查看当前在线用户和在线用户人数。因此所有用户又有列举当前在线用户和当前在线用户人数功能。

4.2.2 用户管理组件的设计

用户管理组件主要是设计视频监控系统中各种用户信息的管理，其属性和接口方法如图 4-1 所示。

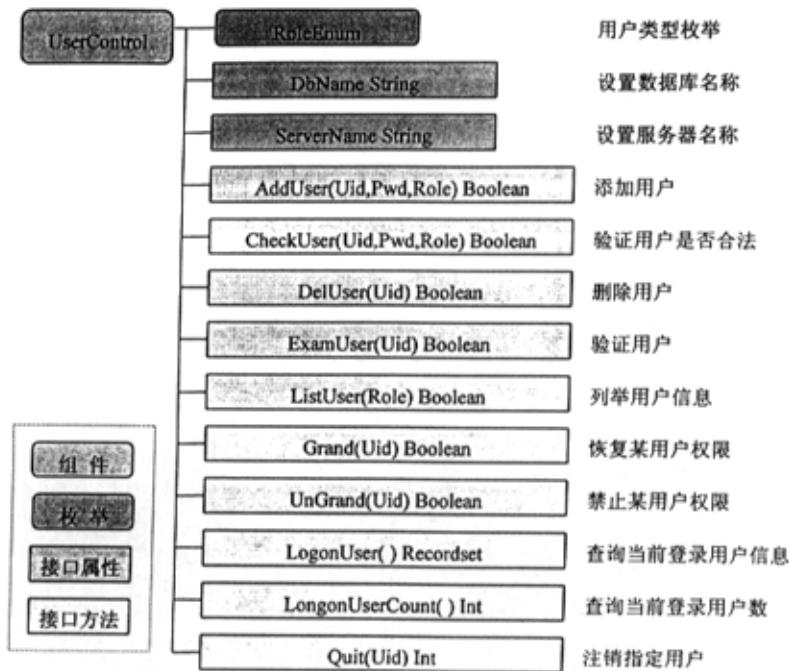


图 4-1 用户管理组件示意图

在组件中，为了充分定位用户类型，使用了枚举类型，分别定义四种用户类型，程序设计如下：

```

Public Enum RoleEnum          '用户权限枚举
    AdministratorEnum = 0   '管理员用户类型
    LeaderEnum = 1          '领导用户类型
    ClientEnum = 2          '一般用户类型
    ForeEnum = 3             '前端用户类型
    NoneEnum = -1           '非法用户类型
End Enum

```

对于创建用户功能，其创建流程如图 4-2 所示。

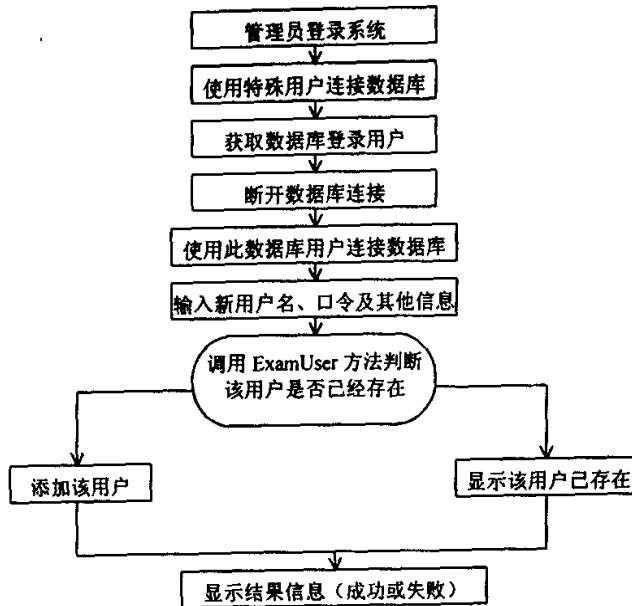


图 4-2 管理员创建用户流程

管理员首先登录系统，系统首先使用特殊用户连接到数据库上，从登录用户与组对应表中判断该用户是否合法，如果合法，则从该表中取出该用户对应的组名，然后根据组名从组与数据库用户对应表中获取对应的数据库用户帐户及口令。再使用该数据库用户帐户连接数据库。在界面中输入新建用户的用户名、口令、权限及其他信息，在界面中直接判断两次口令是否相符。如果所有信息全部输入，组件自动调用自己的接口方法 ExamUser()，判断该用户是否已经存在，如果存在则显示错误信息，表明该用户已经存在；否则将该用户添加到数据库中，显示创建新用户成功信息，创建新用户完成。

创建新用户的程序设计可使用 Visual Basic 或 Visual C++^[27, 30]，下面代码分别使用了这两种语言设计^[4, 9, 38, 50]。

使用 VB 语言设计如下：

```

On Error GoTo Err '如果创建新用户有错，则跳转到 Err 处
Dim bExist As Boolean
bExist = ExamUser(Uid)      '判断此用户是否存在
If bExist = True Then       '如果此用户已经存在，则创建用户失败，退出函数
    AddUser = False
    Exit Function
End If
If Role = AdministratorEnum Or Role = LeaderEnum Or Role = ClientEnum Or Role =
  
```

```
ForeEnum Then
```

```
‘如果添加的用户权限是管理员、领导、一般用户或前端用户，则将该用户添加到数据库中
```

```
    cn.Open ConnectString
```

```
    rs.Open "Users", cn, adOpenForwardOnly, adLockOptimistic, adCmdTable
```

```
    rs.AddNew
```

```
    rs("Name") = Name
```

```
    rs("Power") = Role
```

```
    rs("Uid") = Uid
```

```
    rs("Pwd") = Pwd
```

```
    .....
```

```
    rs.Update
```

```
    rs.Close
```

```
    cn.Close
```

```
    AddUser = True      ‘添加新用户成功，函数返回 TRUE
```

```
Else
```

```
    ‘添加新用户失败，函数返回 FALSE
```

```
    AddUser = False
```

```
End If
```

```
Exit Function
```

```
Err:
```

```
    AddUser = False      ‘如果创建新用户中有任何创建错误，则函数返回 FALSE，
```

```
退出
```

```
End Function
```

使用 VC 语言设计如下：

```
bExist = ExamUser(Uid);
```

```
if(bExist) return FALSE;
```

```
m_pUserSet->Open("Users",
```

```
    theApp.m_pConnection.GetInterfacePtr(),
```

```
    adOpenDynamic,
```

```
    adLockOptimistic,
```

```
    adCmdTable);
```

```
m_pFieldsPtr = m_pUserSet->Fields;
```

```
m_pUserSet->AddNew();
```

```
{
```

```
    m_pFieldPtr = m_pFieldsPtr->Item["NAME"];
```

```
m_pFieldPtr->Value = (_variant_t)Name;
m_pFieldPtr = m_pFieldsPtr->Item["Power"];
m_pFieldPtr->Value = (_variant_t)Role;
m_pFieldPtr = m_pFieldsPtr->Item["Uid"];
m_pFieldPtr->Value = (_variant_t)Uid;
.....
}
m_pUserSet->Update();
m_pUserSet->Close();
return TRUE;
```

对于其他用户功能如删除用户与创建新用户相似，需要注意的是，当删除某个用户时，需要首先判断该用户是否已经登录到系统中，如果已经登录到系统中，则无法删除该用户。

其他用户有验证用户（CheckUser）、注销用户（Quit）、查看当前系统登录用户列表（LogonUser）和登录用户数（LogonUserCount）等接口方法的使用权限。

4.2.3 视频记录管理组件的分析

视频记录管理组件主要是对视频中的各种报警记录和各项操作记录的管理功能。在视频监控中，由于存在各种报警信息，如烟雾报警、红外线报警、温度报警、火警、电流超限报警、电压报警、远动报警和门禁报警等，所以必须对其进行管理。当产生任何一种报警，系统会自动通过视频记录管理组件将这一条报警信息写入到数据库中。通过对各种报警记录的研究，我们可以对这些报警信息的产生原因、产生规律、产生后果等进行分析研究，以防此种情况的再次发生。在监控系统中，任何用户都可以浏览这些报警记录，但只有管理员才可以将某些报警记录删除。

除了报警记录外，监控系统中还包括各种操作记录，如允许报警、禁止报警、定时报警、报警联动、加热开/关、风扇开/关、雨刷开关、浏览报警记录、操作运行记录、系统设置操作、图像调整等，必须要将这些各种操作记录进行存储，以备份各个用户在其登录系统后都进行了那些操作。如果监控出现差错，这可以通过查看以前的操作记录来判断是以前的那些操作引起的，这样就可以很快进行相应的修正。

4.2.4 视频记录管理组件的设计

由于视频记录管理组件主要是对各种报警记录和各种操作记录的管理，所以说该组件设计比较简单，其接口方法如图 4-3 所示。



图 4-3 视频记录管理组件示意图

视频管理组件中包括了一个枚举类型，该枚举是为了定义打开报警记录或操作记录的类型，定义如下：

```
Public Enum OpenRecordEnum '定义打开记录枚举
    OpenAllEnum = 0      '打开所有记录
    OpenNum20Enum = 1   '打开最新的 20 条记录
    OpenNum30Enum = 2   '打开最新的 30 条记录
    OpenNum50Enum = 3   '打开最新的 50 条记录
    OpenNum100Enum = 4  '打开最新的 100 条记录
    OpenDate5Enum = 5   '打开最近 5 天内的记录
    OpenDate7Enum = 6   '打开最近一周内的记录
    OpenDate15Enum = 7  '打开最近半个月内的记录
```

```
OpenDate30Enum = 8 '打开最近 30 天内的记录
```

```
End Enum
```

该组件中有两个属性：数据库名称和服务器名称，可以对属性进行设置，也可以获取这两个属性值。默认情况下，这两个属性的值是从特定的 INI 文件中获取的。

除此以外，该组件中就是各种接口方法，包括增加报警记录、增加操作记录、删除报警记录、删除操作记录、以各种方式打开报警记录或操作记录等。当监控系统自动报警时，就是通过调用视频记录管理组件中的增加报警记录方法将报警信息自动存储到数据库中的；而当前用户所做的操作也是通过调用增加操作记录方法记录下来的。

4.3 两个组件的安全性设计

所有这些 COM+ 组件接口方法的安全性设置都是通过组件服务管理器 (Component Services Administrator, CSA)^[7, 26, 31] 来实现的。也就是说，视频监控系统中使用的 COM+ 安全性都是声明安全性应用方法。

在 CSA 中，首先创建了视频监控系统应用程序 (RdtRMSS 应用程序)，然后将设计好的用户管理组件和视频记录管理组件添加到 RdtRMSS 应用程序中。

在 CSA 中的每一个应用程序都有一个角色 (Roles) 文件夹，在这个文件夹中管理员可以为应用程序创建角色。角色对于一个特定的应用程序和它的组件而言也是明确的。

由于在视频监控系统中包括 4 种用户类型，这在 NT 域中也相应创建了这 4 种用户。在 CSA 的 RdtRMSS 应用程序中，我们也要相应地创建对应于这 4 种用户的角色。创建好这 4 种用户的角色后，就需要将这些角色添加到组件和组件接口中。需要注意的是，用户管理组件中的一些接口方法和视频记录管理组件中的一些接口方法只有管理员才能调用，所以只将管理员角色添加到这些接口方法中。

设置好 RdtRMSS 应用程序的安全性以后，并不意味着 COM+ 会执行这些安全性检测。这需要在 RdtRMSS 应用程序中启动安全性。当启动安全性后，执行 RdtRMSS 应用程序时 COM+ 就会进行安全性检测了。

第 5 章 展示层的设计

Windows DNA 体系结构支持多种展示层类型。展示层有两种基本的类型：“本机用户界面（Native User Interface）”和“基于 Web 的用户界面（Web-based User Interface）”。本机用户界面依赖于由基础操作系统提供的用户界面服务。在 Windows 中，本机用户界面使用 Microsoft Win32 API。基于 Web 的用户界面通常是以 HTML 为基础的，HTML 可以在任何操作系统中利用一个 Web 浏览器来显示。

5.1 基于局域网用户界面的视频监控系统的实现

5.1.1 监控远端系统的实现

监控远端系统主要是针对当前变电所进行监控，如图 5-1 所示。主界面的控制操作分为视频切换、云台镜头控制、设备管理及录像控制等几个部分

1. 视频切换

见主界面右上方按钮组，用户通过界面上的控制按钮选择需要观看的摄像信号。本系统可支持 8, 16, 32 路等多种规模的多媒体矩阵主机，

在该界面中有 12 个按钮，按从左到右，从上到下的顺序编号，每个按钮上标有对应摄像机监视场景的名称；若某路没有安装摄像机则对应钮为“预留”，且变灰，操作无效；最后一个按钮为“翻页”。当摄像机超过 11 个时，以 11 为单位进行分组。这时用户应先按“翻页”钮，翻页到需要切换通道的那一组，再按带摄像机名称钮，则对应监视点的画面出现在屏幕上，同时对应按钮变色。

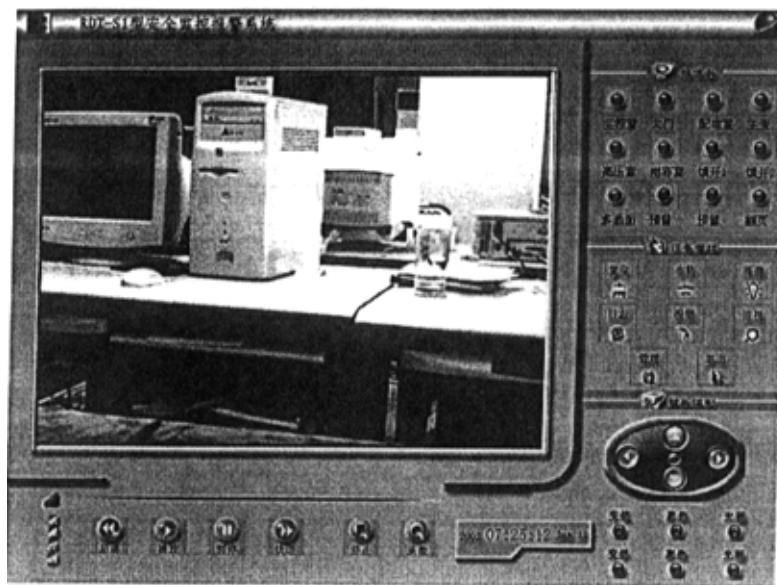


图 5-1 视频监控前端主界面

2. 云台与镜头控制

位于主界面右下方的按钮组。当某路摄像机安装了云台和三可变镜头时，“控制云台”按钮变亮可用，允许用户进行云台与镜头控制，否则变灰不可用。

按“控制云台”钮，弹出“云台与镜头控制”对话框。按下表示“上”、“下”、“左”、“右”四个方向箭头按钮之一，则云台上上、下、左、右转动。松开鼠标左键（抬起按钮），则云台停止转动。

按下“变倍+、变倍-、聚焦+、聚焦-、光圈+、光圈-”钮，可调节三可变电动镜头。

“变倍+”：推近镜头，用于观察远处的物体细节。

“变倍-”：收缩镜头，用于观看场景的全貌。

“聚焦+”、“聚焦-”：调整镜头聚焦，使物体由模糊变清晰。

“光圈+”：使画面由暗变亮。

“光圈-”：使画面由亮变暗。

如果配置单元时，对应解码器上的三个联动开关分配了功能，则在“云台与镜头控制”对话框中可进行控制。

3. 设备管理功能按钮组

设备管理功能主要位于主界面右中部。

(1) “复位”按钮

一旦发生报警，可在变电站前端对报警状态进行手动复位。如停止报警录像，

关闭报警喇叭，向下位矩阵发送预定的报警复位信号等。

(2) “布防”按钮

单击该按钮，将弹出“报警配置”对话框，如图 5-2 所示。

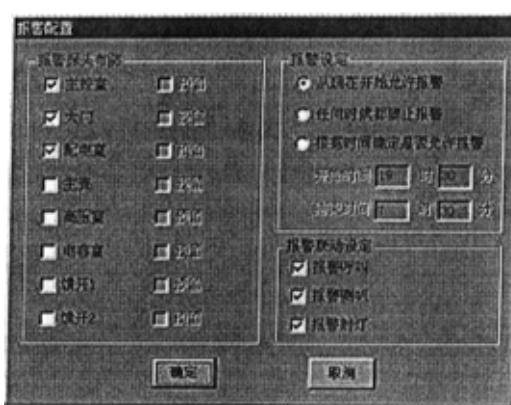


图 5-2 报警配置对话框

该对话框分为三部分，即报警探头布防，报警方式设定和报警联动设定。

报警探头布防列出报警探头安装情况，以复选框的形式设置。选中该路报警探头，则允许报警，否则禁止报警。

报警联动状态设置了报警发生时，是否开启报警射灯，报警喇叭，是否进行当地拨号呼叫等，同样以复选框的形式设置。选中则启用该功能，否则禁止。系统默认的报警联动功能，如视频自动切换到报警区域，打开摄像机电源，开始报警录像和呼叫报警中心灯等，无需在此设置。

报警方式设定有三个单选按钮：

- 1) 从现在开始允许报警；
- 2) 任何时候都禁止报警；
- 3) 根据时间确定是否允许报警，即在设定的时间段内才允许报警。

当选择第二种方式时，布防与联动设定都变灰不可用。而选择第一或第三种方式时，设置的布防与联动状态才起作用。

(3) “图像”按钮

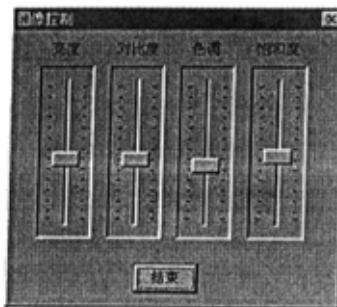


图 5-3 图像控制对话框

系统启动后，有可能不显示图像，也有可能图像显示不清晰，如果不是线路方面的问题，就需要单击“图像”钮，弹出“图像控制”对话框，通过鼠标拖动图中的四个滑块，调整图像卡的四个工作参数来改善图像质量。

德加拉卡的“图像调整”参数：

- 1) 亮度 (Brightness): 可适当调整，如果取值太大，画面苍白。
- 2) 对比度(Contrast): 适当增大该值，可增加图像的层次感。
- 3) 色调(Hue): 一般在缺省位置，否则偏色。
- 4) 色饱和度(Saturation): 一般在缺省位置。适当调高该值，可使图像看起来鲜艳。

(4) “报警”按钮

单击该按钮，则弹出如图 5-4 所示的“报警记录”对话框。该对话框主要用于浏览前端发生报警时所记录的详细的报警信息。在浏览文本信息的同时，操作人员可以用鼠标双击任意一条报警记录，则系统自动回放与该报警相关的视频。同时还提供了功能强大的数据库检索和管理功能。

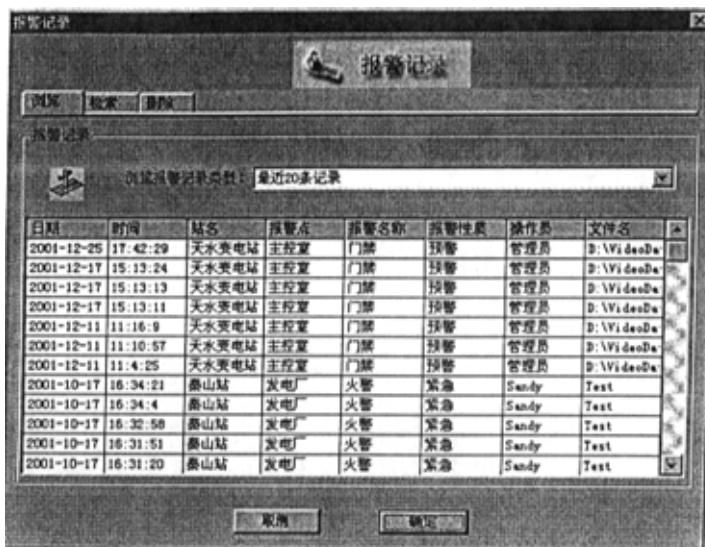


图 5-4 报警记录对话框

程序代码如下：

```

OpenRecordEnum opentype; // 定义打开报警记录类型
opentype = OpenNum20Enum; // 默认为打开最新 20 条报警记录
_RecordsetPtr m_Rs; // 定义 ADO 记录集对象
// 初始化 COM 库, 对组件实例化
HRESULT hResult;
IUnknown* pIUnknown;
hResult = ::CoInitialize(NULL);
if(FAILED(hResult))
{
    ::AfxMessageBox("不能初始化 COM 库!");
    return FALSE;
}
// 创建组件实例
pIUnknown = NULL;
hResult = ::CoCreateInstance(CLSID_RecordShow, NULL,
    CLSCTX_SERVER, IID_IUnknown, (void**)&pIUnknown);
if(FAILED(hResult))
{
    pIUnknown = NULL;
}

```

```
    ::AfxMessageBox("不能创建 RecordShow 对象!");
    return FALSE;
}

//查询接口并使用
if(pIUnknown!=NULL)
{
    hResult=pIUnknown->QueryInterface(IID_RecordShow, (void**)&m_pRecordShow);
    if(FAILED(hResult))
    {
        ::AfxMessageBox("不能获取接口 UserControl 接口!");
        pIUnknown->Release();
        return FALSE;
    }
    m_Rs.CreateInstance(_uuidof(Recordset)); //创建 ADO 记录集对象
    m_Rs = m_pRecordShow->OpenAlarmRecord(openstype);
    //调用组件 RecordShow 接口方法 OpenAlarmRecord, 获取返回的记录集
}

//释放组件
pIUnknown->Release();
pIUnknown = NULL;
::CoUninitialize();
m_AlarmRecordGrid.SetRefDataSource(m_Rs);
m_AlarmRecordGrid.Refresh();
//将获取的记录集显示到 DataGrid 中
```

当用户双击或者选中其中一条报警记录然后单击“确定”按钮，就可以播放该条报警记录的录像视频。

单击“检索”标签页，显示检索页面，可以依照起始时间和终至时间查询，也可以依照站点模糊查询。依照起始时间和终至时间查询是调用视频记录管理组件 RecordShow 的 OpenAlarmRecord2() 接口方法；依照站点模糊查询是调用视频记录管理组件的 OpenAlarmRecord3() 接口方法。

单击“删除”标签页，系统将弹出用户验证对话框，要求只有管理员才能进行这项操作。进入该标签页后，选中一条报警记录后，单击“删除”按钮就可以将该条报警记录删除掉。删除报警记录是调用视频记录管理组件 RecordShow 的 DelAlarmRecord() 接口方法。

(5) “日志”按钮

单击该按钮，则弹出如图 5-5 所示的“运行操作记录”对话框。该对话框主要用于浏览前端操作人员进行设备控制，系统配置等常规的操作信息。为了方便，同时还提供了功能强大的数据库检索和管理功能。



图 5-5 运行操作记录对话框

其“检索”标签页和“删除”标签页与报警记录对话框类似。

(6) “巡视”按钮

点击该按钮，则出现前端电子地图画面。电子地图主要用于直观的显示出变电站平面布局及相关设备的配置。

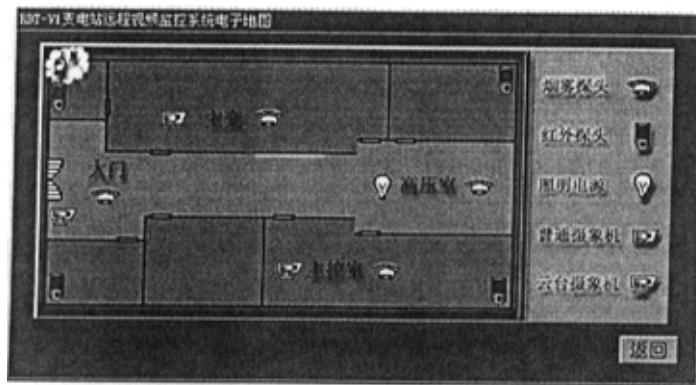


图 5-6 电子地图对话框

单击图中摄像机图标，则显示对应摄像机的视频图像；单击 1 次图中报警探头图标，则允许该路探头报警，再次单击该探头图标，则禁止该路探头报警。

(7) “管理”按钮和“退出”按钮

“管理”按钮用于用户权限管理，对于诸如报警布防，报警记录与操作记录的浏览均需进行登录。

“退出”按钮则用于退出系统运行。退出前须输入有效的口令。

4. 录像回放按钮组

“播放”按钮：弹出视频文件响应目录，双击所要播放的文件即可进行图像回放；

“后退”按钮：播放视频文件时，倒退播放该文件；

“暂停”按钮：暂停正在播放中的视频文件，再次单击，则继续播放；

“快进”按钮：播放视频文件时，加速播放该文件；

“停止”按钮：停止正在播放中的视频文件，回到实时监控画面；

“录像”按钮：对当前监控画面进行手动实时录像，再次单击，则停止录像。

最大手动录像时间可预先设置

5.1.2 监控中心监控系统的实现

监控中心具有监控远端的部分功能（发送控制命令、浏览报警记录和操作记录等）。

监控中心（领导分控）软件安装于报警接收或监控管理中心的计算机上，其主要特点是能够调看和接收前端的图像信号，其界面可采用电子地图方式，形象直观。

用户管理是监控中心的一个重要功能，它可以对视频监控系统中所有的用户进行管理，如图 5-7 所示。

用户管理界面可以查看系统中所有的用户信息，包括用户真实性命，用户名、权限、是否许可、是否在线等信息。可以通过“浏览用户类型”下拉框以查看不同的用户类型信息。

用户管理界面可以创建新用户，输入新用户的信息后，单击“添加”就可以将该用户添加到数据库中。



图 5-7 用户管理界面

添加新用户程序代码如下：

```

BOOL b = FALSE;
//初始化 COM 库, 对组件实例化
HRESULT hResult;
IUnknown* pIUnknown;
hResult = ::CoInitialize(NULL);
if(FAILED(hResult))
{
    ::AfxMessageBox("不能初始化 COM 库!");
    return;
}
//创建组件实例
pIUnknown = NULL;
hResult = ::CoCreateInstance(CLSID_UserControl, NULL,
    CLSCTX_SERVER, IID_IUnknown, (void**)&pIUnknown);
if(FAILED(hResult))
{
    pIUnknown = NULL;
    ::AfxMessageBox("不能创建 UserControl 对象!");
    return;
}

```

```
}

//查询接口并使用
if(pIUnknown!=NULL)
{
    hResult=pIUnknown->QueryInterface(IID_IUserControl,(void**)&m_pUserCo
ntrol);
    if(FAILED(hResult))
    {
        ::AfxMessageBox("不能获取接口 UserControl 接口!");
        pIUnknown->Release();
        return;
    }
    RoleEnum role;
    role=(RoleEnum)m_role;
    b=m_pUserControl->AddUser((_bstr_t)m_name, (_bstr_t)m_uid,
    (_bstr_t)m_pwd,role);
    m_pUserControl->Release();
    m_pUserControl = NULL;
}

//注销
pIUnknown->Release();
pIUnknown = NULL;
::CoUninitialize();
if (b==FALSE)
    MessageBox("添加用户失败!", "添加用户信息", MB_ICONERROR|MB_OK);
else
    MessageBox("成功添加该用户!", "添加用户信息", MB_ICONASTERISK|MB_OK);
```

在“删除用户”标签页可以删除指定的用户，其实是调用用户管理组件 UserControl 的 DelUser() 接口方法；“设置用户许可权限”标签页可以对指定用户的许可权限进行设置，其实是调用用户管理组件 UserControl 的 Grand() 和 UnGrand() 接口方法。

为了统一，用于领导和各部门和远程分控，现在也采用监控软件来担当。区别在于：作为分控的用户，在配置“通信中心”时，应分配较低的用户权限。而应设定监控中心具有最高权限。最大的区别是：在“通信中心”配置文件中 Alarm_CenterIP 的值为监控中心所在 PC 机的 IP 地址，这样当报警发生时，通信中

心只把报警信息传递到监控中心。

1. 通信中心

通信中心主要包括三个部分：

(1) 监控前端列表 显示与本“通信中心”相连的所有监控前端单位的名称、通讯地址、当前所处的状态等信息。

(2) 用户列表 显示在本“通信中心”注册的所有合法用户名，其当前的工作状态，是否正在与某个前端通信，是正常结束还是异常结束等。

(3) 运行记录 列表显示软件运行以来，最近 100 个运行记录出处。

因为“通信中心”软件，需要连续不断地处于运行状态，为了防止无关人员随意关闭该软件，所以系统设计了口令保护功能。每次退出“通信中心”软件都要求操作人员输入口令，核实正确后，才能退出。

2. 领导分控

领导分控主要特点是能够调看和接收前端的图像信号，其界面可采用电子地图方式，形象直观。

监控中采用的电子地图分为两种：主地图与分地图。主地图只有一个，用于显示监控前端的地理分布。分地图可有若干个，用于显示监控前端各监视点的分布位置。

每个地图数据由两个文件组成：一个是地图的图形信息文件，目前使用 BMP 格式；另一个是热点信息文件，其扩展名为 INF。这两个文件的文件名相同。如：成都供电段接触网电子地图的两个文件为：成都供电段接触网.BMP 和成都供电段接触网.INF。

目前热点分为三大类：(1) 摄像机 (2) 报警探头 (3) 继电开关，INF 文件中按这个顺序存放各热点的描述信息。为了用图标准确反映热点的特征，又把热点细分为以下几种类型：

1—代表普通摄像机

2—带云台，三可变镜头摄像机

3—防盗报警探头（双鉴，红外等）

4—烟雾探头

5—温度探头

6—应急按钮

7—继电开关

8—报警喇叭

9—照明设备

每个热点的描述格式如下：热点在 BMP 图上的 X 坐标，Y 坐标，热点具体类型，热点的名称。例如：hot1=134, 54, 2, 主控制室，说明“主控制室”热点为带云台控

制的摄像机热点，布于地图上坐标为（134，54）处。

主电子地图上可以表示热点是否在通信；点电子地图上标有摄像机的热点可进行视频切换；可以对带云台与镜头控制的摄像机进行控制；可以自动存储报警录像；也可以对报警允许进行设定等。

5.1.3 安全性设计

1. 操作密码保护机制

系统采用了严格的操作密码保护机制：(1) 对无人值班系统中的现场监控部分，由于正常情况下无人操作，因此专门设立了“操作保护”功能，防止无关人员对系统的随意操作。(2) 通过网络对远端站点的访问有两级保护机制，首先对远端访问的用户必须在通信前进行注册；其次，开始通信时需要输入用户口令，口令正确方可登录，而且某个时刻只允许用户登录一次，即不能重复登录。

2. 防火墙

供电段的计算机网络数据库中存在大量的敏感数据，所以需要为供电段的计算机网络系统系统安装防火墙防止非法人员窃取信息。防火墙是放在多个网络之间的系统或系统组合，通常，防火墙是一组硬件设备-路由器、主计算机，或者是路由器、计算机和配有适当软件的网络设备的多种组合。逻辑上，防火墙是分量器、限制器和分析器。防火墙的作用是保护网络在阻止非授权用户的访问敏感数据的同时允许合法用户无妨碍的访问网络资源。具体的讲，防火墙可以减少外部侵袭者突破内部网络系统的可能性，也能阻止用户发送危险的信息（未加密的文件）。

3. 其他

系统中采用了 Windows NT 的域管理器来管理局域网中的各个用户，这样就可以将视频监控系统的安全性和 Windows NT 的安全性联系起来。

5.2 基于 Web 用户界面的视频监控系统的实现

5.2.1 Web 服务器的设计

如果希望在 Internet 网上对变电所进行监控，或者说是领导如果不在监控中心和监控远端，他希望可以查看到某个变电所的监控情况，那么他可以通过 Internet 网来进行监控。

在监控中心，我们部署了一个 WEB 服务器，在 WEB 服务器上安装了 Microsoft

公司的 IIS (Internet Information Services, Internet 信息服务) [23]。首先在 IIS 中新建一个 Web 站点, 命名为 RdtRMSS, 设定 TCP 端口号为 90。因为系统默认的 TCP 端口号为 80, 可能会被其他 Web 站点使用, 所以可以设定一个没有使用过的端口号就可以了。需要注意的是, 如果 Web 站点的 TCP 端口号为 80, 在访问该站点时就只需要输入站点的 IP 地址就可以了; 如果 Web 站点的端口号不是 80, 则访问此站点时就需要在 IP 地址后加上端口号, 如: <http://192.168.0.120:90/>。设置 RdtRMSS 站点的主目录为设计好的 Web 页面主目录, 默认文档为该站点的起始网页文件, 如 Default.asp。

IIS 是通过 Internet 服务管理器 (Internet Services Manager, ISM) 管理的, 当然也可以一组 HTML 页面进行管理, 这些页面和 ISM 都能提供远程管理功能, 差别在于 ISM 必须要安装在远程计算机上, 而 HTML 管理页面只要求远程计算机安装浏览器。

HTML 管理页面为远程管理 IIS 提供真正简便的方式, 而且更快、更有效, 如图 5-8 所示。



图 5-8 HTML 管理页面

在打开 HTML 管理页面时, 必须要知道 Web 服务器的管理 Web 站点的端口号。在安装 IIS 时, 安装程序会自动产生一个介于 1000~9999 之间的随机端口号, 并将之分配给安装 HTML 管理页面时所创建的管理 Web 站点 (Administration Web Site)。

这个初步的安全措施可以防止不知道端口号的人员的访问。从远程的计算机上访问 HTML 管理页面，必须知道这个端口号。这个端口号可以通过 IIS 中管理 Web 站点的属性对话框中获取。缺省时，只有安装了 Web 服务器上的浏览器才能够调用 HTML 管理页面，因为在 IIS 中系统设置了 IP 限制。如果希望使用远程计算机来管理，这可以在限制列表中增加自己的 IP 地址。该地址可以是局域网上远程计算机的 IP 地址，也可以是代理服务器或 ISP 的 IP 地址（如果通过网络连接），如图 5-9 所示。

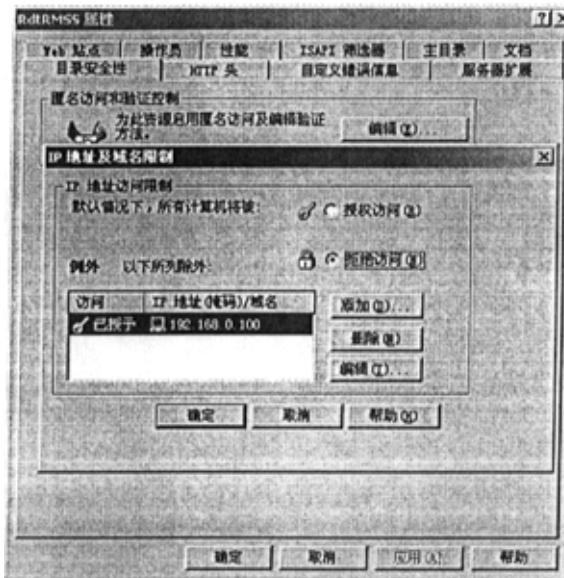


图 5-9 设置 Web 服务器访问的 IP 限制及域名限制

当然，IP 限制也可以设置为一组计算机，输入网络标识及子网掩码；如果是域名限制，则需要输入域名以决定是授权访问还是拒绝访问。

当我们设置好 IP 限制和域名限制后，远程管理 IIS 只能在这些允许访问的机器上进行。

在本机上，使用 IIS 的 Internet 服务管理器中管理 Web 站点来管理是比较方便的。

5.2.2 客户端的实现

在 Internet 网上进行视频监控，实行实时的控制，实现起来比较困难。在视频监控系统的 Web 客户端，主要是进行一些查看浏览操作，包括一些简单的用户管

理，报警记录、操作记录的浏览以及以前报警录像的查看。由于在 Internet 网上传输视频数据需要很宽的带宽，所以我们可以对视频数据进行压缩，网页上播放工具可以使用 Windows Media Player。

当用户打开主页面时，需要用户登录，系统通过用户管理组件进行判定该用户是否合法。如果合法该用户才可以对其他功能进行操作。由于 Web 客户端的用户（为一般用户）只能对信息具有查看功能，所以安全性相对来说比较高。

Web 客户端主页面如图 5-10 所示。



图 5-10 Web 客户端主页面

该页面中显示出了 Web 主页面及打开报警记录信息页面的情况。

浏览报警纪录代码如下：

```
<%
Dim objRdtRMSS,rsAlarmRecord,i
Set objRdtRMSS = Server.CreateObject("RdtRMSS.RecordShow")
'创建视频记录管理组件
If objRdtRMSS Is Nothing Then    '如果创建错误，则显示错误信息！
    Response.Write "..."
Else
    Set rsAlarmRecord = Server.CreateObject("ADODB.Recordset")
    Set rsAlarmRecord = objRdtRMSS.OpenAlarmRecord(OpenType)
    '将当前类型的报警记录返回给指定ADODB的Recordset上
    While Not rsAlarmRecord.EOF
```

```
..... '显示当前报警记录  
rsAlarmRecord.MoveNext  
WEnd  
Set rsAlarmRecord = Nothing  
Set objHello = Nothing      '清除组件对象  
End if %>
```

可以从显示的报警记录中获取要查看的报警录像文件，然后将其打开进行查看。操作操作与打开报警记录操作类似。由于在 Web 客户端，只能接受视频数据而不能发送特殊的控制命令，所以其功能相对单一。WEB 服务器发送什么视频数据，该客户端就接受什么数据。对于用户管理，由于 Internet 网安全性较差，所以不提供创新用户、删除用户和设置用户权限功能。

5.2.3 安全性设计

Internet 的安全问题，主要是黑客用各种方法截取基本的数据包数据，使用“不好”的口令访问系统，等等。尽管这种威胁看来势不可挡，但可以采取一些方法来保护安全结构设计较差的网络系统。管理员必须主动地研究所用产品的 Internet 安全特性。如果做好预防工作，就能在 Internet 上构建一个安全的系统。

我们通过如下方法增强了 Web 服务器的安全性：

1. 硬盘格式 我们将 Web 服务器格式化为 NTFS 格式，因为 NTFS 提供了比 FAT 或 FAT32 更多的安全特性。所以我们选择使用了 Windows 2000 支持的 NTFS 的最新版本（NTFS5）。

2. 禁止匿名的网络访问 我们修改了 Web 服务器上注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\restrictanonymous 的值由 0 改为 1，可以禁止匿名的网络访问。

3. 禁止自动共享网络共享 由于 Windows 2000 允许共享创建的网络驱动器并自动共享所有的驱动器，并在 C:\Winnt 目录下创建了一个 ADMIN\$ 共享。为了保证服务器的安全，我们在注册表中新建下列键来禁止这项功能。在注册表路径 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters 下新建项 AutoShareServer，值设置为 1，可以禁止自动共享网络共享。

4. 禁止注册表对远程访问的许可 由于 Windows 2000 支持对注册表的远程访问，所以运行 Regedt32 打开注册表编辑器，选中注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg，然后选择菜单“安全”/“权限”命令，打开“winreg

的权限”对话框，可以看到管理员具有完全的控制，Backup Operator（备份操作员）有读访问权限。由于在该 Web 服务器上使用了其他备份软件，所以我们可以删除这个读访问权限。

5. 将管理员的帐户改名 因为系统管理员对系统有很大的权限，所以黑客总是试着以管理员的身份攻击系统。为了防止这种情况的发生，我们将 Web 服务器上的管理员帐号进行了更改。同时也建立了一个假帐户为 Administrator，并设置该帐户没有任何访问权限。

6. 禁止使用管理员工具 系统中有些工具只有管理员才能使用。为了保护这些工具，我们在 Web 服务器上创建了一个新的文件夹 C:\Admin_Tools，将这些工具全部拷贝到该文件夹内，并设置文件夹的许可权限，使得只能由管理员帐号才可以使用这些工具。这些工具包括有：cmd.exe, cscript.exe, ftp.exe, net.exe, telnet.exe, telnetc.exe, wscript.exe 等。

7. 防火墙和代理服务器 防火墙是能够在两个网络之间，通常是 Internet 和内部网络之间，建立一道安全的屏障。代理服务器是特殊类型的防火墙，能够代表一个计算机完成其自身不能或不想完成的工作。防火墙和代理服务器的用途是取得不可路由的地址，并把它们转换成可路由的地址。由于它们能控制经过它所传递得信息，所以对于防止自己网络受到侵害和骚扰有很大作用。另外，防火墙也可以阻止向 Internet 上传递敏感信息。

在视频监控系统中，由于在 Internet 网上，我们只需要用户进行查看和发一些简单控制命令，对一些敏感的控制命令和操作就需要屏蔽掉或者用户直接无法进行这些操作，这样就可以比较安全地保护服务器上的数据。

结 论

目前，国内外对视频监控系统都有了较多研究，并有一些成熟系统在各领域得到成功应用，如电信无人值守通信机房的环境监控；电力系统变电站、机房、电厂设备管理及监控；高速公路交通监控系统；智能大厦及重要地点安保系统等等。对于铁路牵引供电系统，目前随着铁路现代化的发展以及企业减员增效压力的增加，对铁路变（配）电所（分区所、开闭所）采用综合自动化系统和实现无人值班的要求日益迫切，因此，逐步采用远程视频监控系统是铁路发展的必然趋势。但是，与其他视频监控系统不一样，由于铁路变（配）电所是沿铁路线分布的，与其相关的通讯信道也是沿线分布的，且各变电所之间相距较远（一般几十公里），一方面，通道受系统独有的电气化干扰严重，另一方面，信道相对比较紧张，因此铁路的通信信道多采用总线型或环状引入。这就要求系统必须能适应铁路不同的通道结构。正因如此，目前还没有一套成熟的远程视频监控系统在铁路牵引供电系统得到成功应用。本文结合铁道部科技计划项目—牵引变电所安全监控及综合自动化成套技术的研究，在对铁路远程视频监控系统功能分析的基础上，提出了一种分布式远程视频监控系统（DRMMS）方案，并对系统组成、系统设计等进行了详细分析。经过初步实验，得出如下结论：

1. 利用 Windows DNA 技术来设计视频监控系统，使系统具有很好的扩充性和扩展性。
2. 采用 COM 组件设计视频监控系统的商务逻辑层，使系统更易于升级换代。
3. 利用三层结构的方式来设计视频监控系统，对各层都进行安全保护，可以极大地提高系统的安全性。
4. 通过将视频监控系统中的商业逻辑设计到 COM 组件中，增加了系统的商业秘密保护。

通过对论文的撰写，对视频监控系统的分析设计，可以更加深入地了解了 SQL Server 内部的安全结构；使用面向对象编程，可以从整体上把握系统，对系统整体构架的设计起到了很好的作用；对 COM 组件的具体设计以及其安全机制的设置，可以保护商业秘密，同时可以使客户端尽可能的瘦；还有 UML 技术、视频压缩解压技术的学习和应用等，这些都使我获得了很大的收益。

Windows DNA 无论是在技术上、软件开发的模式上来说，还是实际应用来说都有更多的优点。但是由于视频监控系统实在庞大，本论文只给出了其整体结构分析和部分程序设计，其它很多模块都没有具体设计完成。

此外，由于篇幅的限制，系统中如 ER 图转换层数据库、使用 Rational Rose 实

现系统分析以及监控前端用户界面的设计等都逐一写出。

当然系统的功能还有很多需要进一步完善和扩充，例如视频的传输和视频压缩解压算法的选择应该根据具体情况而定等。

致 谢

本论文是在导师彭强副教授的精心指导下完成地，它凝聚着导师对我的悉心培养和教导所付出的大量心血。在研究生期间，导师严谨的治学作风、忘我的工作精神给我留下了深刻的印象。这些优秀的品质让我受益终身，使我明白了许多做人治学的道理，端正了我的学习和工作态度。在此表示最衷心的感谢，并对几年来彭强老师在学习和生活上的关心与帮助表示衷心的谢意。

衷心感谢电气学院的陈维荣教授、计算机学院的诸昌铃教授、马永强教授等对我的悉心指导和帮助；最后对在学习及各方面中给予我帮助和支持的其他老师、同学和朋友谨此表示深深的感谢！！

参考文献

- [1] [美]Mary Kirtland, 北京博彦科技发展有限公司译. 基于组件的应用程序设计. 北京大学出版社, 1999.
- [2] [美]Ash Rofail, Yasser Shohoud 著, 邱仲潘等译. COM 与 COM+从入门到精通. 电子工业出版社, 2000.
- [3] 潘爱民. COM 原理与应用. 清华大学出版社, 2000.
- [4] [美]Jim Maloney. Visual C++ 6 DCOM 开发指南. 北京: 清华大学出版社, 2000.
- [5] [美]Frank E. Redmond III. Building Windows DNA Applications. <http://msdn.microsoft.com/library/techart/windnadesign.htm>.
- [6] [瑞典]Sten Sundblad, Per Sundblad 著, 前导工作室译. Windows DNA 可扩展设计. 机械工业出版社, 2001.
- [7] [美]Richard C.Leinecker 著, 高智勇, 赵崑等译. COM+技术大全. 机械工业出版社, 2001.
- [8] [美]John Paul Mueller 著, 刘建军, 卢刚等译. Visual Basic COM+编程宝典. 电子工业出版社, 2001.
- [9] 张龙祥. UML 与系统分析设计. 人民邮电出版社, 2001.
- [10] 汤庸. 多媒体数据库与网络应用. 人民邮电出版社, 2000.
- [11] 蔡安妮等. 多媒体通信技术基础. 电子工业出版社, 2000.
- [12] F. Kuo 等. 多媒体通信协议与应用. 清华大学出版社, 1998.
- [13] 毕厚杰. 多媒体信息的传输与处理. 人民邮电出版社, 1999.
- [14] 王卫安. 用 Client/Server 数据库存储 GIS 图文数据. 微电脑应用, 2001.
- [15] Murat Tekalp. Digital Video Processing. 清华大学出版社, 1998.
- [16] 钱清泉. 电气化铁道微机监控技术. 中国铁道出版社, 2000.
- [17] 余松煜, 张文军, 孙军. 现代图像信息压缩技术. 科学出版社, 1998.
- [18] Murat Tekalp. Digital Video Processing. 清华大学出版社, 1998.
- [19] Video coding for Low Bitrate Communication. ITU-T Recommendation H. 263 Version 2, 1998.
- [20] Y. Wang and Q. Zhu. "Error control and concealment for video communication: A review". *Proceeding of IEEE*, vol. 86, pp. 974-997, May 1998.
- [21] *Signal Processing: Image Commu.* (Special Issue on Error Resilience and Robustness for Video Coding Technology), vol. 14, may 1999.

- [22] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Transactions on consumer electronics*, vol. 39, pp. 455-460, August 1993.
- [23] [美]Richard Anderson Chris Blexrud 等著, 刘富太, 张立民等译. ASP 高级编程. 机械工业出版社, 2000
- [24] Don Box 著, 潘爱民译. COM 本质论. 中国电力出版社, 2001
- [25] [美]约翰·斯万科著, 徐颖译. COM 编程精彩实例. 中国电力出版社, 2001
- [26] [美]Randy Abernethy 著, 江浩等译. COM/DCOM 技术内幕. 电子工业出版社, 2000
- [27] [美]Tom Armstrong, Ron Patton 著, 董梁, 丁杰等译. ATL 开发指南(第二版). 电子工业出版社, 2000
- [28] Mark Priestley. Practical Object-Oriented Design With UML. McGraw-Hill Companies, Inc., 2000
- [29] Eugene Kain 著, 健莲科技译. MFC 经典问答. 中国电力出版社, 2001
- [30] Brent Rector, Chris Sells 著, 潘爱民, 新语译. 深入解析 ATL. 中国电力出版社, 2001
- [31] Dale Rogerson 著, 杨秀章译. COM 技术内幕. 清华大学出版社, 1999
- [32] Stanley B.Lippman 著, 侯捷译. 深入探讨 C++ 对象模型. 华中科技大学出版社, 2001
- [33] Stanley B.Lippman 著, 侯捷译. Essential C++ 中文版. 科技大学出版社, 2001
- [33] Scott Meyers 著, 侯捷译. Effective C++ 中文版 2nd Edition. 科技大学出版社, 2001
- [34] Mary Kirtland. COM+ 程序设计模型 .
<http://www.zdnet.com.cn/developer/rescenter/story/0,2000081634,20004290-1,00.htm>
- [35] Bruce Shankle. Find Recombinant Success with Windows DNA.
<http://msdn.microsoft.com/library/en-us/dnmind99/html/dna.asp?frame=true>
- [36] Frank E. Redmond III. Introduction to Designing and Building Windows DNA Applications.
http://msdn.microsoft.com/library/en-us/dndna/html/windnadesign_intro.asp?frame=true
- [37] Brian Hart. Managing Favorites Using Windows DNA.
<http://msdn.microsoft.com/library/en-us/dncdev00/html/vc00j1.asp?frame=true>
- [38] Ari Bixhorn. Microsoft Visual Basic for Applications and Windows DNA 2000.
<http://msdn.microsoft.com/library/en-us/dndna/html/vba4dna.asp?frame=true>
- [39] 开发者 : 技术 . Microsoft 多层平台的演变 .
<http://www.zdnet.com.cn/developer/resources/story/0,2000081634,20003647,00.htm>

- [40] 开发者：技术. Microsoft. N 层开发模型.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20003775-1,00.htm>
- [41] Mike Schelstrate. Stress Testing Data Access Components in Windows DNA
Applications.
<http://msdn.microsoft.com/library/en-us/dnmdac/html/dnastress.asp?frame=true>
- [42] Frank E. Redmond III. Guide to Designing and Building Windows DNA
Applications.
http://msdn.microsoft.com/library/en-us/dndna/html/windnadesign_guide.asp?frame=true
- [43] Stephen Rauch. Windows DNA: Building Windows Applications for the Internet
Age.
http://msdn.microsoft.com/library/en-us/dndna/html/msdn_windnapps.asp?frame=true
- [44] Ken Spencer. Windows DNA Then and Now.
<http://msdn.microsoft.com/library/en-us/dnmind99/html/beyond1299.asp?frame=true>
- [45] 开发者：技术. Windows DNA 应用程序数据访问组件的强度测试.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20002320,00.htm>
- [46] 开发者：技术. 从 Hello World 学习 Windows DNA.
<http://www.zdnet.com.cn/developer/study/story/0,2000081626,20003869,00.htm>
- [47] 开发者：技术. 设计和创建 Windows DNA 应用程序简介.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20000258,00.htm>
- [48] 开发者：技术. 什么是 Windows DNA.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20000249,00.htm>
- [49] 开发者：技术. 主要的 windows DNA 性能错误及其预防.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20002506,00.htm>
- [50] 开发者：技术. 用 VB6 开发 ADO 2.0 中间层数据访问组件.
<http://www.zdnet.com.cn/developer/tech/story/0,2000081602,20004671,00.htm>
- [51] 董保东. COM 组件实现应用程序前后台安全型设计. 计算机科学. 2001, 28 卷
6 月(增刊)
- [52] 袁鹏飞. SQL Server 7.0 数据库系统管理与应用开发. 人民邮电出版社, 2000

攻读硕士学位期间发表的论文

发表论文：

董保东. COM 组件实现应用程序前后台安全型设计. 计算机科学. 2001, 28 卷 6
月 (增刊)

编写书籍：

董保东. Flash 4.0 网页动画教室. 四川大学出版社, 2000

董保东. 多媒体精彩范例设计 Authorware 5.2&Photoshop 6.0 现场实作. 北京希
望电子出版社, 2001