

摘要

随着计算机技术和网络技术发展，楼宇智能化技术在我国各类建筑中得到了广泛应用。然而传统的楼宇控制技术仅限于楼宇内部的集中控制，而无法实现楼宇外的控制与管理，这对于全国很多楼宇物业企业,或者想用极少的人力管理成本实现多个楼宇的集中控制的小公司来说具有很大的挑战性。本论文就是基于这种要求,探讨基于 Web 远程集中控制与各楼宇客户端集中控制相结合，实现不同楼宇间的远程管理与控制。

在 Web 远程控制与楼宇设备集中控制管理模式中，通过信息交换、共享和集中方式，将各个具有完整功能的独立分系统组合成一个有机的整体，实现功能集成、网络集成、软件界面集成以及信息一体化集成。论文首先对研究过程中涉及的关键技术进行了详细综述，这些技术包括 LonWorks 现场总线技术、SQL Server 数据库技术、JAVA 技术、C/S+B/S 组网络、.Net 技术。其中重点介绍了 LonWorks 现场总线技术、数据库技术和.Net 技术，通过 LonWorks 现场总线技术实现楼宇中各个网元设备的集中控制与数据通信，通过数据库技术实现对监控数据的集中管理与数据查询,然后把数据库中的监控数据通过.java 技术、.Net 技术进行页面展示。在功能上主要实现集中监视、联动控制和管理、分布智能，分散控制、信息共享与信息查询等功能。

论文在整个系统设计的过程中通过集成楼宇中各种子系统，把它们统一在单一的操作平台上进行管理。采用 C/S+B/S 组网络方式和采用数据库服务层、WEB 服务器、应用服务器实现三层结构的设计思路,实现楼宇远程与本地集中控制，使楼宇中各种弱电系统的操作与管理更为简易，更有效率，对于楼宇远程集中控制、节约人力成本起到积极的意义。

关键词： BAS，模块，LonWorks 现场总线，楼宇自动化

ABSTRACT

The minimal cost of human resources management to achieve centralized control of multiple buildings in the country. This thesis is based on such a request to explore centralized Web-based remote control with the respective building client centralized control and management is the combination of remote management and control between the different buildings.

In order to achieve through the exchange of information sharing and centralized, each with a fully functional independent subsystems into an organic whole, to achieve the automation of system maintenance and management, linkage to coordinate the operational capabilities, functional integration, network integration, software interface integration and information the Integration. Firstly, a detailed overview of the key technologies involved in the research process, these technologies include LonWorks field bus technology, the SQL Server database technology, JAVA technology, the C / SB / in group S network, Net technology. Which focuses on the LonWorks field bus technology, database technology. Net technology, LonWorks field bus technology to achieve centralized control of building equipment of various network elements and data communications and database technology to achieve centralized management of monitoring data and data query, and then monitoring data in the database through. java technology, Net technology page impressions. Followed by the commencement of the design of the specific details on the design requirements of the system analyzed in detail and analyzed in detail, focusing on building centralized control and monitoring, focus on monitoring, linkage control and management of distributed intelligence, distributed control, information sharing information query functions. Detailed design and analysis of various functional modules, which work processes, networking and related needs. Detailed test for the various functional modules of the system, including the login test, system performance analysis and testing, system testing and adjustment of the final system design requirements.

Papers in the overall system design process through the various subsystems, integrated buildings to be unified in a single platform to manage. C / SB / S-set of

network and database services layer, WEB server, application server to achieve the three-tier structure of the design ideas, building a remote centralized control with local, more simple operation and management of buildings in a variety of weak systems and more efficient for building remote centralized control, saving labor costs to the positive significance.

Keywords: the BAS, module, LonWorks fieldbus, of METASYS, building automation

目 录

| | |
|-------------------------------------|-----------|
| 第一章 绪论 | 1 |
| 1.1 研究背景和意义 | 1 |
| 1.2 研究现状 | 1 |
| 1.3 内容与组织结构 | 3 |
| 第二章 需求分析与开发工具 | 5 |
| 2.1 系统设计目标 | 5 |
| 2.2 系统需求分析 | 6 |
| 2.2.1 业务需求分析 | 6 |
| 2.2.2 功能需求分析 | 6 |
| 2.2.3 系统结构分析 | 9 |
| 2.3 系统开发技术 | 12 |
| 2.3 .1LonWorks 技术 | 12 |
| 2.3.1.1 LonWorks 技术概述 | 12 |
| 2.3.1.2 LonWorks 技术的硬件组成 | 13 |
| 2.3.1.3 LonWorks 通信协议 LonTalk | 16 |
| 2.3.2 .Net 技术 | 19 |
| 2.3.2.1 .NET 与 Web Services..... | 19 |
| 2.3.2.2 ADO.NET 数据库访问技术 | 22 |
| 2.3.3 B/S 与 C/S 结构 | 24 |
| 2.3.3.1 C/S 结构 | 24 |
| 2.3.3.2 B/S 结构 | 25 |
| 2.3.3.3 C/S 和 B/S 比较 | 26 |
| 2.4 本章小结 | 26 |
| 第三章 系统详细设计 | 27 |
| 3.1 系统结构详细设计 | 27 |
| 3.1.1 Web 结构设计 | 27 |
| 3.1.1.1Web 服务层设计 | 28 |
| 3.1.1.2 Web 数据处理层设计 | 29 |

| | |
|------------------------------|-----------|
| 3.1.2 基层结构设计 | 32 |
| 3.1.2.1 Lon 网络集成 | 32 |
| 3.1.2.2 现场网络与上位机通信实现技术 | 33 |
| 3.2 系统功能设计 | 35 |
| 3.2.1 系统总体结构设计 | 35 |
| 3.2.2 停车管理子系统 | 35 |
| 3.2.3 自动化系统设计 | 37 |
| 3.2.4 消防报警管理子系统设计 | 38 |
| 3.2.5 综合保安管理系统的应用 | 39 |
| 3.3 数据库设计 | 40 |
| 3.3.1 数据库逻辑结构(E-R) | 40 |
| 3.3.2 数据结构与程序的关系 | 45 |
| 3.3.3 数据库具体设计 | 46 |
| 3.4 本章小结 | 47 |
| 第四章 系统功能实现..... | 48 |
| 4.1 消防预警实现 | 48 |
| 4.1.1 底层温度信息采集 | 48 |
| 4.1.2 消防预警软件实现 | 49 |
| 4.2 系统功能页面展示 | 53 |
| 4.2.1 系统运行日志界面 | 53 |
| 4.2.2 自动化系统配置界面 | 53 |
| 4.2.3 报警管理排表界面 | 54 |
| 4.2.4 消防预警界面 | 55 |
| 4.2.5 综合保安管理界面 | 56 |
| 4.2.6 车库收费管理 | 57 |
| 4.3 消防预警温度信息 Web 查询 | 58 |
| 4.4 本章小结 | 61 |
| 第五章 系统运行测试..... | 62 |
| 5.0 测试概述 | 62 |
| 5.1 测试环境 | 62 |
| 5.2 测试场景与工具 | 63 |
| 5.3 测试结果分析 | 64 |

| | |
|---------------------------------|-----------|
| 5.3.1 分析摘要 | 64 |
| 5.3.2 运行 Vuse | 66 |
| 5.3.3 平均事务响应时间 | 66 |
| 5.3.4 运行 Vuser-平均事务响应时间 | 67 |
| 5.3.5 每秒点击数 | 67 |
| 5.3.6 每秒点击数-每秒 HTTP 请求响应数 | 68 |
| 5.3.7 吞吐量-每秒点击数 | 69 |
| 5.3.8 事务摘要 | 69 |
| 5.3.9 事务性能摘要 | 70 |
| 5.3.10 Windows 资源 | 70 |
| 5.4 总结性能测试结论 | 71 |
| 5.5 本章总结 | 71 |
| 第六章 总结和展望 | 72 |
| 致 谢 | 73 |
| 参考文献 | 74 |

第一章 绪论

1.1 研究背景和意义

在经济高速发展的今天，全球楼宇管理部门一直为寻找更舒适、便捷、优雅楼宇环境而努力着。为实现这一目标，他们找到了通过科学技术与建筑技术相结合的解决方法，通过高科技实现室内环境的智能化管理。而智能楼宇就是以这种目标而提出的解决方案，它结合了计算机技术（Computer）、通信技术（Commission）控制技术（Control）、与建筑技术（Architecture），即 3C+A 解决方案^[1]。它是通过科学技术实现环境与自然的和谐发展。

楼宇自控系统 BAS(Building Automation System)是智能楼宇中重要的组成部分，它为建筑系统提供一个安全、舒适、便利的室内环境，通过智能楼宇控制系统进行楼宇内电器设备的自动化控制与管理。实现楼内能源最低消耗控制、楼内电机设备运行状态自动检测，例如：控制空调运行、制冷机温度、自动排水、楼道视频监控、消防控制、安全告警灯，通过自动化控制系统进行统一管理与调度^[2]。

信息技术和网络技术的发展是楼宇智能化的物质基础，在新的技术环境下，对楼宇智能化系统、服务提出了新的要求。自动化控制系统必须能为楼宇提供设备运行状态检测、设备节能控制、消防安全等，并能根据预先设定的规则进行安全控制（能实现自主启停、控制风速等）、图形处理、告警、节能节排等。

然而，目前对于楼宇控制技术在智能化楼宇中的应用仅仅是实现楼宇内部的集中控制，而无法实现楼宇楼宇外的控制与管理，这对于在全国有很多楼宇物业企业或者想用极少的人力管理成本实现多个楼宇的集中控制带来了很大的困难。本论文就是基于这种要求下，探讨基于 Web 远程集中控制与各自楼宇客户端集中控制管理方式相结合，实现不同楼宇间的远程管理与控制。

1.2 研究现状

20 世纪 90 年代，随着世界范围内信息高速公路的迅速发展，装备了由计算机控制的楼宇自动化控制系统的建筑，满足了信息时代下人们对生活和工作的需要，建筑体系实现了 BAS 综合管理，楼内的空调、供水、防火、防盗、变配电系统等

均由计算机控制，使人们真正感受到了舒适、方便、安全。同时，楼宇自动化控制系统也使得建筑体系本身的功能得到了全新的提升，深入了人们工作、生活、娱乐、文化等各个领域，这样也大大推动了智能建筑的步伐，并做为信息高速公路的主要节点，充分体现了现代高科技信息技术给人类带来的进步。目前，有楼宇自动化控制系统的智能建筑在世界各国正雨后春笋般的出现。据不完全统计，到 20 世纪末，美国新建和改建的办公楼约有 70% 是智能建筑，总数共计上万座。

国内智能化楼宇建设开始于 1990 年的北京的发展大厦，随后逐步出现一大批新智能楼宇，到 2003 年底共建成 3500 幢智能楼宇，且智能楼宇建设开始渗透到公共场所（例如：博物馆、展览馆、机场等）和企业楼宇中，影响着人们的日常生活，智能楼宇为人们提供了更舒适、更安全、更便捷。2003 年智能化楼宇总投资 16 亿，其中自动化控制占了一亿，据专业部门预测，21 世纪世界上将有一半以上智能建筑兴建在中国^[3]。

智能建筑系统是不同独立系统综合而成的大系统，计算机通信网络是各个系统连接的神经系统，它实现楼宇内的各个物理设备关联、统一，在该网络中实现语音、数据、图像、视频数据的高速传输。楼宇自动化系统作为整个建筑物的核心子系统，网络技术在楼宇智能化技术中的应用主要就体现在这一子系统中，因此，下面主要围绕楼宇自动化这一子系统展开探讨^[4]。

（1）集散型控制系统（DCS）在楼宇自动化系统中的应用

集散型计算机控制系统(DCCS)又名分布式计算机控制系统，简称集散型控制系统(DCS)。该技术主要实质是通过计算机技术实现集中管理分散控制方式，在架构上采用了三级架构模式：控制、操作、仪表站。DCS 结合了计算机技术、信号处理技术、测量控制技术、网络通信技术和人机接口技术。产品主要有西门子公司的 S600A POGEE 楼宇自控系统。但传统的 DCS 系统存在（1）必须采用 I/O 模拟量信号。（2）无法进行远程设备控制与调整。（3）系统采用封闭式管理，系统无法自由扩展^[5]。

（2）现场总线技术在楼宇自动化系统中的应用及研究

现场总线控制系统 FCS，是用于现场仪表与控制系统和控制室之间的一种全分散、全数字化、智能、双向、多变量、多点、多站的通信系统。它具有：开放、互操作性强、设备自动化程度高、整体结构高度分散等特点。目前现场总线类型主要有 LonWorks 现场总线、EIB 现场总线等，在底层通信协议上有楼宇自动控制网络通信协议 BACnet 技术。

支持 LonWorks 技术主要产品有如霍尼伟尔公司的 EBI 系统、西门子公司的

APOGEE 系统、江森公司的 METASYS 系统等。EIB 现场总线产品主要有如 ABB i-bus EIB 系统^[6]。

(3) FCS 和 DCS 综合技术在楼宇自动化系统中的应用

FCS 采用的是全分布式控制，采用采集各个智能设备节点数据的管理模式。这种模式主要缺点是，设备多的情况下容易出现网络数据传输拥塞的问题。DCS 弥补了 FCS 上缺陷，它具有信息化、集成化、结构开放、支持多种现场总线控制技术。在结构上采用管理层、自动化层、现场层三个层次、管理层主要是采用网络协议实现楼宇设备信息采集，自动化层主要是实现通过总线数据通信协议技术实现对 DDC 分站控制器进行控制；现场层主要是现场 DDC 控制器控制与管理。

(4) 以太网络技术在楼宇自动化系统中的应用及研究

以太网络技术是指充分利用以太网络成本低、资源丰富、固定 IP 实现远程管理等特点进行楼宇设备管理。通过 Ethernet 总线技术实现了从管理层到现场设备控制层通信协议的兼容和统一,主要产品有北京楼宇自动化中心开发的基于以太网的 ENC-20011P 智能建筑测控系统^[7]。不过以太网络技术的缺点是开发基于以太网的控制系统产品的难度较大，这样带来了整体成本的升高，同时在实时性、可靠性有待进一步提高。

1.3 内容与组织结构

论文首先对介绍楼宇自动化的概念，分析集散系统 DCS、现场总线技术 FCS、工业以太网等计算机网络技术在楼宇自动化系统中的应用、研究现状及优缺点。讨论几种现场总线在楼宇自动化系统中的应用现状，分析存在的问题。然后引出智能楼宇管理系统(IBMS)的主要目的-集成楼宇中各种子系统，把它们统一在单一的操作平台上进行管理。在论文中采用 C/S+B/S 方式结合以及采用数据库服务层、WEB 服务器、应用服务器实现三层结构的设计思路。通过 Ethernet (或 Internet) 将各种功能的控制器（如：RCU、SDC-II、VCM、ACM 等）与通讯、数据服务器相联，完成空调设备、机电设备、制冷系统、给排水系统和电梯系统等的监视及控制，用户还通过基于 Web 方式的图形界面可了解建筑物内部设备运行的实时数据、告警信息、实时分析曲线和历史数据记录；可直观显示设备平面布局、状况及故障部位；并可实时监视设备现场的网络视频图像等。同时用户端图形界面可由用户自由组态，并以网页站点的方式向外发布，用户无论在本地或远端，只需透过 WEB 浏览器即可随时随地进行设备监控，这些旨在让楼宇中各种弱

电系统(ELV)的操作更为简易，更有效率。

论文在智能楼宇管理系统设计时使用到了 PCT 相关技术、VBScript 技术实现客户端页面，以及采用 JAVA 技术、ACTIVE-X 技术等技术实现 Web 页面功能展现。而在建筑物中的设备统一接入到控制网络中使用了 LonWorks 现场总线技术技术进行底层网元设备统一控制。

论文共分为七章，具体如下：

第一章为绪论。主要介绍了论文产生的背景，以及目前楼宇自动化研究现状，同时介绍了本论文设计结构。

第二章为研究中的关键技术。主要介绍了系统设计中所用到的技术，主要是在智能节点中使用 LonWorks 现场总线技术方法，以及 Web Services, Net 相关技术。

第三章为需求分析。对系统设计进行需求分析，主要从系统设计原则、系统框架设计分析、系统功能分析等角度进行需求分析。

第四章为系统功能详细设计。主要介绍了系统功能详细设计过程以及代码开发、数据库设计等。

第五章为系统功能实现。主要进行了系统功能实现效果展示。

第六章为系统运行测试。主要通过单元测试、系统性能测试等方面进行系统运行测试。

第六章为总结与展望。对系统设计进行总结与展望。

第二章 需求分析与开发工具

2.1 系统设计目标

总体设计目标是：以计算机网络为基础，以软件工程为核心，通过信息交换、共享和集中，将各个具有完整功能的独立分系统组合成一个有机的整体，实现系统维护和管理的自动化、联动协调运行能力、详细的管理功能以及策略应用开发平台，彻底实现功能集成、网络集成、软件界面集成以及信息一体化集成。系统集成的实施效果是，确保大楼内所有设备处于高效、节能、最佳运行状态，提供一个安全、舒适、快捷的工作环境。具体可分解为如下子目标^[22]：

(1) 集中监视、联动控制和管理

系统可对各子系统进行集中统一式监视和管理，将各集成子系统的信息统一存储、显示和管理在同一平台上，并为其他信息系统提供数据访问接口。重点是要准确、全面地反映各子系统运行状态。并能提供建筑物关键场所的各子系统综合运行报告。

以各集成子系统的状态参数为基础，实现各子系统之间的相关策略联动，实现本系统作为智能策略应用开发的强大平台，从而更进一步实现智能建筑的人性化管理、舒适、节能、安全和便利的目标。

(2) 分布智能，分散控制

各子系统进行分散式控制保持各子系统的相对独立性，以分离故障、分散风险、便于管理。

(3) 标准化

系统设计及其实施严格按照国家建设部和地方的有关标准进行，所选用的系统，设备，产品和软件符合工业标准和主流模式，同时针对项目需求的特殊性，支持多种通信协议，以在不增加额外成本的情况下灵活集成诸家厂商设备和系统。

(4) 信息共享与信息查询功能及其前瞻性

充分考虑到智能建筑技术、电子信息技术及软件技术的快速发展，系统设计在技术上采用国际通用的 CMM 模型规范研发流程，保证软件的整体质量，同时在一些成熟的关键技术领先国内乃至世界的先进水平，并代表业界的发展方向。

(5) 系统开放性和接口通用性

系统采用开放式的模块化架构，具有很好的可扩充性和保护已有投资的升级性。采用不增加额外成本的情况下可无缝集成诸家厂商设备和系统的接口技术。

(6) 安全性和可靠性

安全和可靠是对智能建筑的基本要求，是系统工程设计所追求的主要目标之一。采用业界领先的分布智能技术，保证智能建筑在应急状态下的正常运行。

(7) 管理人性化

从需求分析到程序的界面设计，实现专业的技术性管理向人性化管理转变。人性化还应充分体现在智能建筑使用的高效、舒适、便利和安全等方面。以及还必须考虑到人机界面友好、系统运行稳定、便于维护和扩充修改

2.2 系统需求分析

2.2.1 业务需求分析

智能楼宇自动化控制管理系统的目的是集成楼宇中各种子系统，把它们统一在单一的操作平台上进行管理。通过系统集成，构筑整个建筑物的中央监控与管理界面，通过可视化的、统一的图形界面，进行便捷的对多个子系统进行管理、实时监视、控制和集中的统一管理，与直观展示系统设备的运行状态，并可实现跨子系统的联动功能，当子系统设备发生故障报警时，集成管理系统应当直观形象的显示其设备位置和报警信息，提供管理人员进行处理，并可以执行相关的联动动作。

同时为更好的实现对各子系统的集成，系统应选用基于分系统平等方式的集成管理架构，考虑到大楼设备将来高度集成化和高点击率的需要，采用基于 WEB 浏览，支持 OPC、XML 方式，可以实现跨平台管理的集成管理系统。

2.2.2 功能需求分析

(1) 总体功能描述

在本论文设计中总共分为 5 个子系统（自动化系统、消防报警系统、停车管理系统、综合保安管理系统、远程 Web 信息管理系统），如图 2-1 所示（，每个子系统实现的功能各不一样。

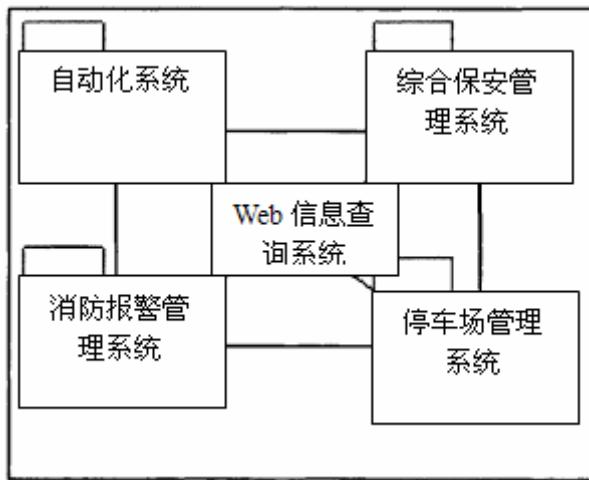


图 2-1 系统功能结构

智能楼宇自动化集成管理系统通过分布式网络将各子系统的管理集成到同一个计算机支撑平台上，建立起整个办公大楼建筑的中央监控与管理界面。通过一个可视化的、统一的、基于 WEB 浏览方式的图形窗口界面，管理员人员可以十分方便、快捷地对建筑物内所有子系统进行实时监视、控制和集中统一管理。从而提升系统运作效率。

智能集成系统需实现以下多方面的功能集成：

- 1) 建立统一的实时数据库，实现 BAS、CAS、FAS、SAS 等子系统的无缝集成，通过中央站完成建筑物内的设备监控、管理、数据采集、联动控制和图像监视等功能；
- 2) 对楼内机电设备进行数据处理，提供其运行曲线、时间累计、报表。
- 3) 提供有关办公楼能源消耗统计、设备运行及维护指南、设备安装位置及数量等相关数据，供管理者查询、打印。
- 4) 提供综合查询功能。可迅速检索某一信息点所在区域平面图，也可通过图形名称来检索该图形中若干信息点；以时间检索报警点状态；以传感器或探测器类型检索状态；以时间检索持卡人进出控制通道记录。
- 5) 提供各类专门报告书。如按分区提供电能用量信息，作为财务核算的依据，为管理提供第一手资料。
- 6) 提供远程通信接口，管理者可通过远地节点与集成系统主机的通信，对新大楼进行远程监控。
- 7) 能够设定系统操作员的操作密码、操作级别、软件操作权限、设备控制权。系统对进入系统的操作人员，以本人操作密码的方式鉴别和管理。系统只赋予操

作员进行同级别的数据访问及修改权和设备控制权。

8) 设备发生故障或者有安防报警时，操作界面将自动弹出报警画面，显示报警地点和设备，并伴有声光报警通知相关管理人员。

9) 提供弱电系统联动的逻辑，在突发事件发生时，能自动为决策者提供指挥的操作界面，并按编制的程序联动相关子系统处理事故。

10) 集成系统能与 MS-SQL 数据库连接，使 BMS 系统数据库能为中心的管理服务；

11) 系统应与计算机信息网络无缝集成，支持 INTERNET/INTRANET 连接，支持 WEB 浏览方式，支持远程访问；

12) 系统应该是一个开放的集成系统，支持 OPC、BACnet、TCP/IP、LONmark、DDE 动态数据库交换等标准协议；

13) 系统中央工作站应配置数据库访问接口，以便与 FAS、SAS 等子系统集成；同时系统应配置实时数据库、满足业主监控要求。

(2) 自动化子系统

自动化系统作为楼宇智能化系统中重要组成部分，它包含了楼宇电力系统、空调系统、照明系统、排水系统、电梯系统等子系统，子系统间相互联系、相互通信，在本设计中就是通过 Lon 现场总线进行数据通信与集中控制。

其结构如 3-2 所示

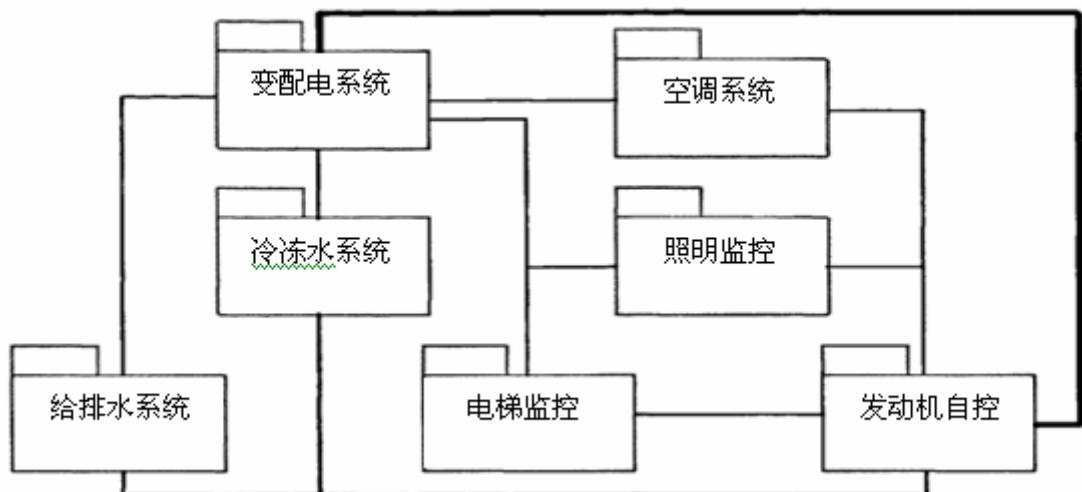


图 2-2 自动化系统结构图

(3) 综合保安管理子系统

综合保安管理子系统主要包含防盗报警系统、闭路电视监控系统、门禁系统、

电子巡更系统。其结构如 3-3 所示

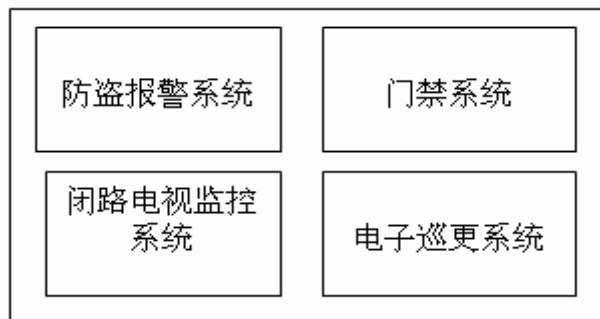


图 2-3 综合保安管理系统

(4) 消防报警管理系统

消防报警管理子系统主要是楼宇设备自动系统触发广播通告、火警通告等子系统，它主要包含：温度探测器系统和烟雾探测器系统，其结构如 3-4 所示。



图 2-4 消防报警管理系统

(5) 停车场管理子系统

该模块主要包括停车车位信息、设备控制信息、收费系统等，其组件如图 3-5 所示。

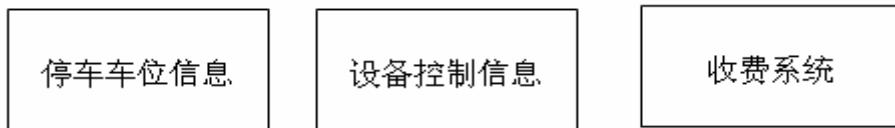


图 2-5 停车管理系统

2.2.3 系统结构分析

(1) 系统总体解决方案结构图如图 2-6 所示

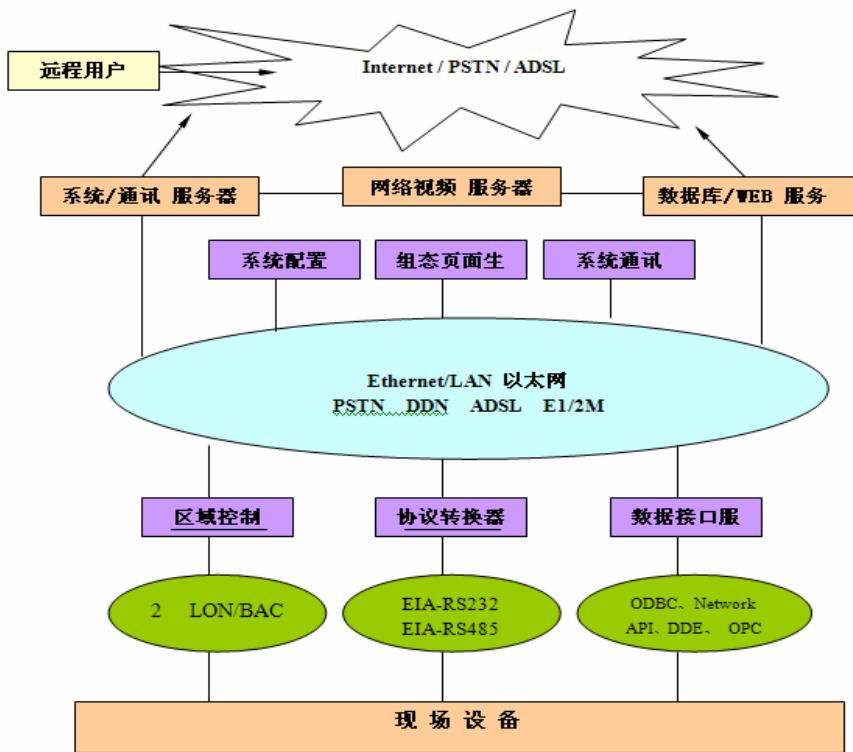


图 2-6 系统解决方案结构图

系统利用最新互联网络技术，采用先进技术将 4C 技术（即 Computer 计算机技术、Control 控制技术、Communication 通信技术、CRT 图形显示技术）有机地结合，以实现建筑物内部信息的采集和综合、信息的分析和处理、信息的交互与共享。同时采用基于计算机互联网络（Internet/Ethernet）及嵌入式（Windows CE）技术的设备管理系统。该系统为用户提供全新“物件导向设计”功能、大量的通用控制函数和算法、丰富的预设置设备控制点模板等。用户通过基于 Web 方式的图形界面可了解建筑物内部设备运行的实时数据、告警信息、实时分析曲线和历史数据记录；可直观显示设备平面布局、状况及故障部位；并可实时监视设备现场的网络视频图像等。同时用户端图形界面可由用户自由组态，并以网页站点的方式向外发布，用户无论在本地或远端，只需透过 WEB 浏览器即可随时随地进行设备监控。在统一管理与控制方面智能楼宇管理系统具有集中监控、集中维护、集中管理、分散控制、远程管理控制和维护等功能。系统能够根据人们的需要，自动调节建筑物内的各种机电设备，包括电梯系统、空调系统、照明系统、CCTV、安防 IC-CARD 门禁系统、停车场管理系统及消防系统等，以创造适合不同需求的人们所要求的不同的舒适工作环境，提高工作效率。

(2) 系统总体层次结构

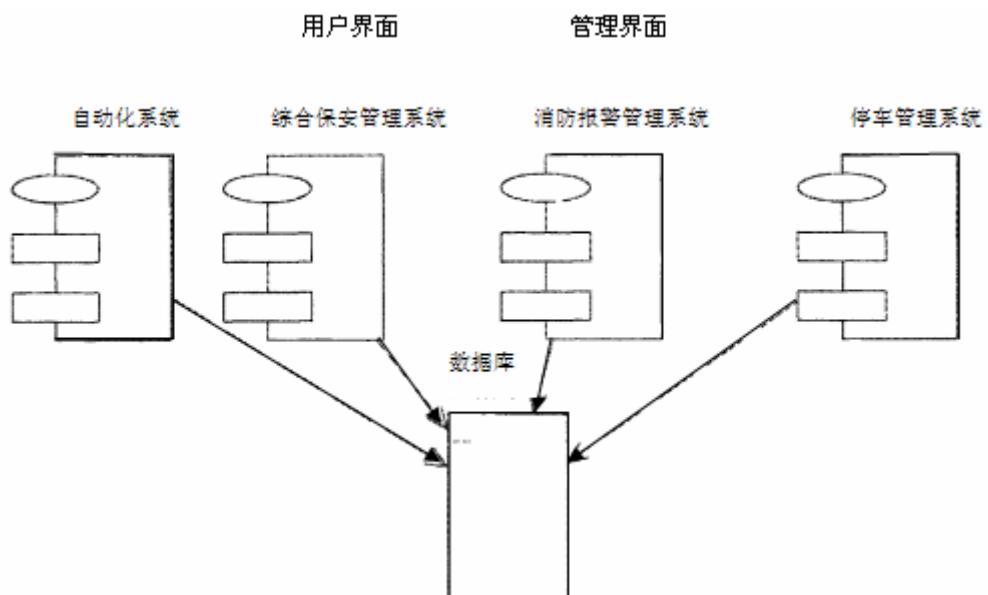


图 2-7 软件总体结构图

(3) 楼宇 Web 信息查询模块

根据我们对楼宇监控系统业务的理解，在降低模块间耦合度的原则下，其网络结构为：

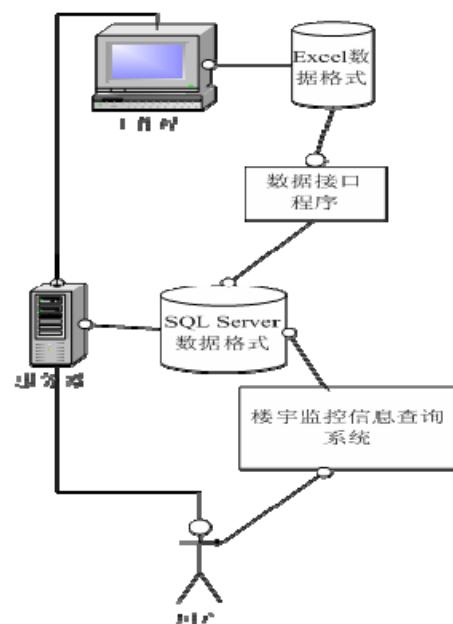


图 2-8 楼宇信息查询网络结构

(4) 火灾自动报警系统原理

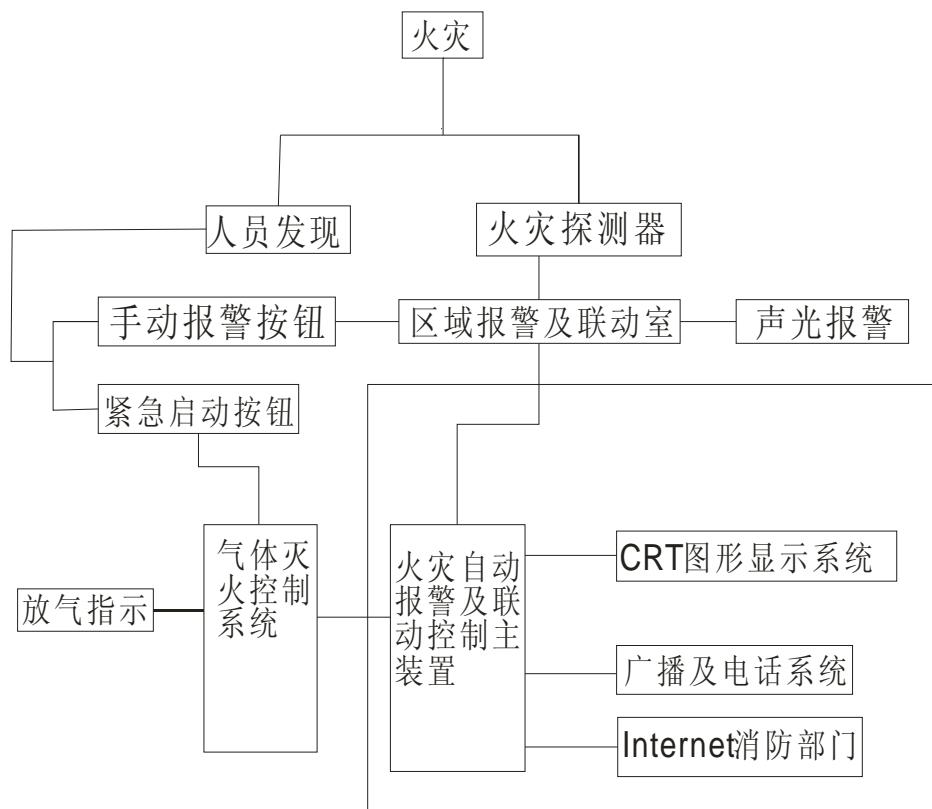


图 2-9 火灾自动报警系统原理图

2.3 系统开发技术

2.3.1 LonWorks 技术

2.3.1.1 LonWorks 技术概述

Lon 总线是一项局部操作网络, 它支持开放式通信协议 LonTalk 以及 OS1 的七层协议, 通过网络变量实现设备节点间数据传递, 其核心是神经元芯片(Neuron chip), 神经元芯片主要负责整个 Lon 总线网络的通信处理、数据采集与控制, 同时整个网络充分考虑到了未来的可扩展性, 在互通性、兼容性上做了充分的考虑, 实现了不同厂家、不同设备间互通性问题, 这也是 Lon 现场总线得以在智能化控制管理中广泛应用的原因。LonWorks 技术包括以下几个组成部分^[8]:

- A.LonWorks 通信控制处理器—神经元芯片(Neuron Chip);
- B.LonWorks 的通信协议—LonTalk;
- C.LonWorks 面向对象编程语言—Neuron C;

D.LonWorks 技术是用于开发监控网络。

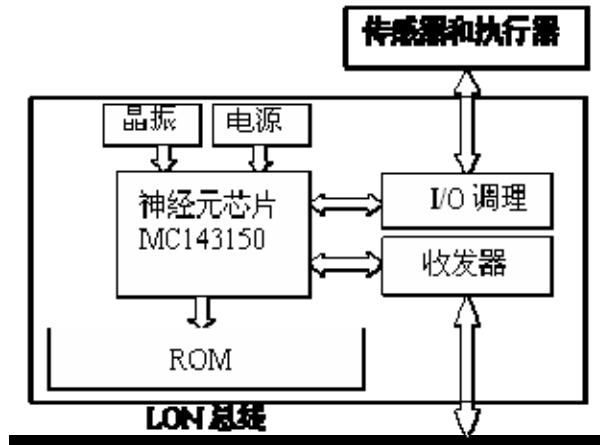


图 2-10 典型的 LON 节点方框图

2.3.1.2 LonWorks 技术的硬件组成

LonWorks 技术具有其它现场总线具有的所有优点，是一个完整的技术平台，在 Lon 网络中，硬件包括：智能节点、神经元芯片、各种传感器、控制设备、数据收发器、电源，LonWorks 就是实现了这些硬件间的数据通信。LonWorks 在软件技术内容包括：支持 1S0/0S1 的七层模型协议、监控网络的设计、开发套件等。典型的 Long 网络结构如图 2-1 所示^[9]。

(1) 神经元芯片(通信控制处理器)

LonWorks 技术的核心是神经元芯片，神经元芯片是一个集和通信、控制、调度和 I/O 功能的高级 VLSI 器件，通过硬件和固件的有机结合，实现 LON 网络节点所需要所有关键功能：处理所有 LonTalk 通信协议消息、传感信号输入和控制信号输出，存储和安装指定的参数及程序，实现各种应用功能等。

表 2-1 神经元芯片硬件配置比较表

| | MC143150 | MC143120 | MC143120E2 | TMPN3120E2 |
|------------|----------|----------|------------|------------|
| 处理器 | 3 | 3 | 3 | 3 |
| RAM | 2048 | 1024 | 2048 | 1024 |
| ROM | | 10240 | 10240 | 10240 |
| EEPROM | 512 | 512 | 2048 | 1024 |
| 16 定时器/计数器 | 2 | 2 | 2 | 2 |
| 外部存储器接口 | 有 | 无 | 无 | 无 |
| 管脚 | 64 | 32 | 32 | 32 |

神经元芯片型号主要是 3150 和 3120。3150 信号芯片优点是支持外部存储器，

其使用范围是大型复杂的系统以及需要软件扩展环境；而 3120 芯片因其带有内部 ROM 存储设备，里面已经集成了小型系统，适用于低成本、小型控制、不需要外部存储的环境。其各自硬件资源配置情况见表 2-1。从各自硬件配置信息中可以看出：3150 多了外部数据线和地址线及少量控制线的引脚，支持以外接 ERPOM，而 3120 则不行；两种设备都有内部 EEPROM，主要存储 网络配置、编址信息、以及固化在硬件上用户应用程序代码和只读数据；同时芯片上静态 RAM 中(K2 和 IK)，设有数据堆栈，可以存储系统数据、系统缓冲数据^[10]。

(2) 处理单元

如图 2-2 所示，神经元芯片包含三个微处理器：媒体访问控制、网络和应用处理器。

A. 媒体访问控制处理器：

媒体访问控制处理器负责介质访问控制，通过网络缓冲区进行和网络处理器进行数据通信与传递，以及通过驱动通信子系统的硬件完成碰撞回避算法控制。

B. 网络处理器

网络处理器主要负责处理网络变量进程、编址、处理事项进程、报文鉴定、软件定时器、网络管理和路由等功能。通过网络缓冲区与媒体访问控制处理器、应用处理器通信处理。

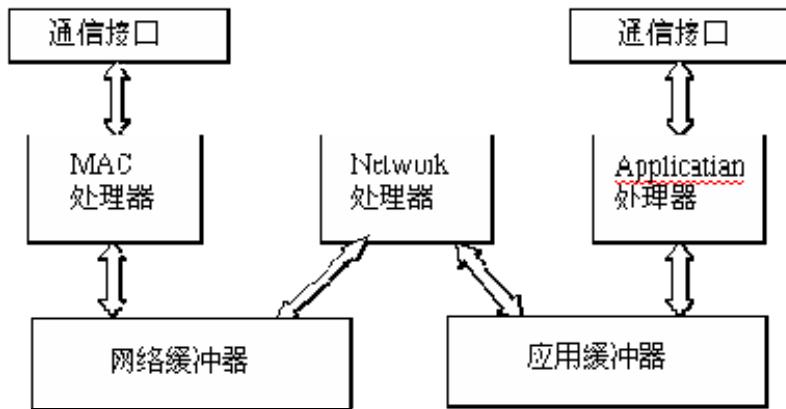


图 2-11 神经元芯片的三个处理器

C. 应用处理器：

应用处理器主要负责执行用户代码，通过共享存储器中的应用缓冲区与网络处理器通信。该层属于应用层，在网络应用层，已经屏蔽了底层物理固件编写，用户只需要通过简单的函数调用就可以实现新功能扩展。而底层处理都交给媒体访问控制处理器和网络处理器进行处理。

(3)存储器

MC3150 存储器包括：512 字节的 EERPMO 用来存放网络配置和地址表，48 位神经元功码，2048 字节的 RMA 包含堆栈段、应用程序和系统程序的数据区、LonTalk 协议的应用缓冲区和网络缓冲区。

(4)输入输出

神经元芯片的三个处理器通过 16 位地址总线和 8 位数据总线与片内存储器、网络通讯接口、定时/计数器、I/O 口驱动电路（I00-I010）相连，其内部结构如图 2-3 所示。在芯片编程上通过函数 io_in() 和 io_out() 进行 I/O 操作。

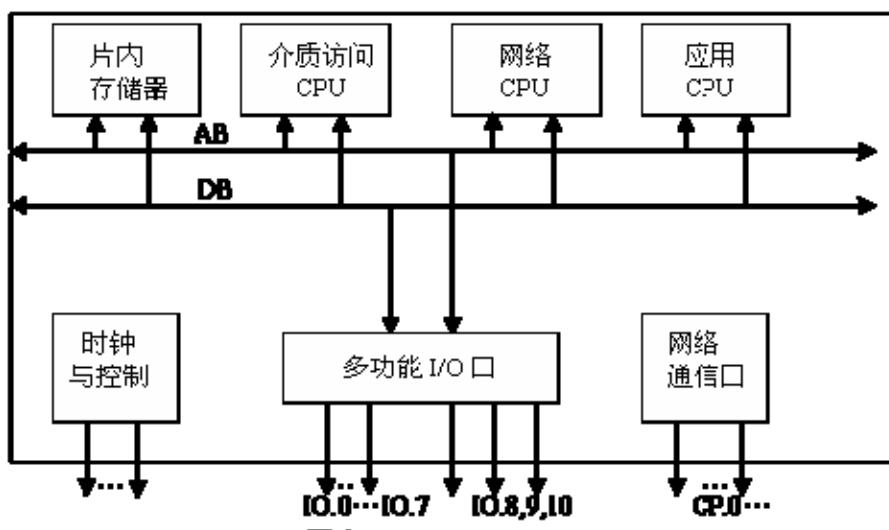


图 2-12 MC143150 芯片内部结构

(5)LonWorks 通信介质

Lon 总线的一个非常重要的特点是它对多种通信介质的支持。使用最广泛的是双绞线，其次是电力线，其它包括无线、红外、光纤、同轴电缆等。下表 2-2 列出了几种典型的收发器^[11]。

表 2-2 Neuron 芯片的典型收发器类型

| 收发器 | 波特率 |
|--------------------------|--------------------------|
| EIA — 485 | 300bps — 1.25bps |
| 自由拓扑和总线型双绞线 | 78kbps — 1.25mbps |
| 电力线（载波） | 4kbps |
| 电力线（扩频） | 10kbps |
| 无线（200MHZ、450MHZ、900MHZ） | 1200bps, 4800bps, 78kbps |
| 红外 | 78kbps |
| 光纤 | 1.25mbps |
| 同轴电缆 | 1.25mbps |

Neuron 芯片可以通过调整五个通信管脚接口模式(单端、差分、专用模式)、编码方案、波特率调整实现不同介质连接扩展，单端模式可用于光纤、同轴电缆或无线射频收发器，差分模式用于连接带隔离变压器的双绞线网络收发器，专用模式用户设计的智能型网络收发器连接。这些模式用户可以根据具体环境进行模式调整，提高了通信方式的灵活性。

Neuron 芯片可支持多种通信介质和多种网络拓扑，其中双绞线以其高的性能价格比而应用最为普遍，如图 2-4 所示。

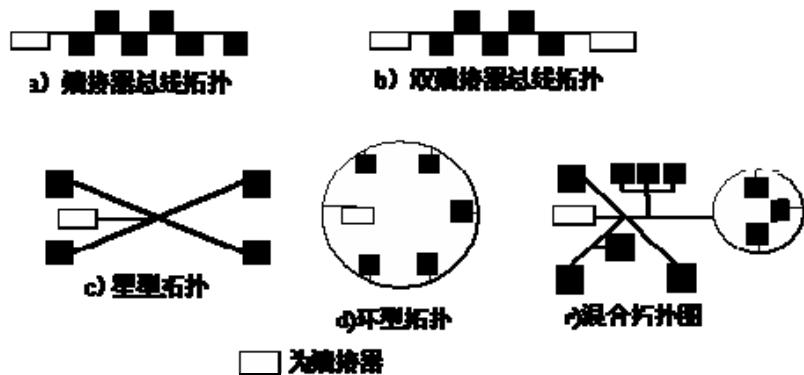


图 2-13 网络拓扑类型

(6) 路由器

智能节点间数据通信需要一个物理信道，而路由器就是负责连接两个信道之间的 LonTalk 信息和控制两个信道之间数据包传送，其路由器负责内容如：

- A. 扩充通道的容量。可以通过扩充路由器网络的容量实现智能节点间收发器的数据传输长度和负载。
- B. 连接不同的通信介质或波特率。通过路由器设置实现通道与物理介质间综合管理与通讯。
- C. 提高 Lon 总线的可靠性，全面提高网络性能。可以通过路由器信道安全管理控制，实现不同信道间的安全隔离，但不会影响其信道间原来工作机制和内容。

2.3.1.3 LonWorks 通信协议 LonTalk

在 Lon 网络中，LonWorks 技术使用 LonTalk 协议。LonTalk 协议完全支持开放系统互连(OSI)模型，支持 7 层网络协议模型，这也是 Lon 现场总线和其它总线不同的地方。在物理介质支持上 LonTalk 协议支持包括双绞线、电力线、无线、红外线、同轴电缆和光纤。

(1) LonTalk 协议的特点^[12]

LonTalk 协议是具有以下特点：发送短报文数据；网络上的节点采用单片机；通信带宽不高；多节点，多通信介质；可靠性，实时性。

(2) LonTalk 的七层协议

LonTalk 包容了 Lon 总线中包含网络操作系统，通过固件网络开发协议实现各种介质间的可靠数据传输。在网络实现中在面对对象通信时，只需要网络变量修改方式就实现所有网络通信的功能。LonTalk 和 OSI 的七层协议的比较如表 2-3 所示。

表 2-3 LonTalk 和 OSI 的七层协议的比较

| OSI 层次 | 标准服务 | LON 提供的服务 | 处理器 |
|--------|----------|--------------------|---------|
| 应用层 | 网络应用 | 标准网络变量类型 | 应用处理器 |
| 表示层 | 数据表示 | 网络变量、外部帧 | 网络处理器 |
| 会话层 | 远程遥控操作 | 请求/响应，认证 | 网络处理器 |
| 传输层 | 端对端的可靠传输 | 应答，非应答，点对点，广播 | 网络处理器 |
| 网络层 | 传输分组 | 地址，路由 | 网络处理器 |
| 链路层 | LLC 子层 | 帧结构，数据解码，CRC 错误校验 | MAC 处理器 |
| | MAC 子层 | P 预测 CSMA，碰撞规避，优先级 | MAC 处理器 |
| 物理层 | 电路连接 | 介质，电气接口 | MAC 处理器 |

(3) LonTalk 协议的寻址方式和网络地址结构

LonTalk 地址包括使用域、子网、节点地址。通过分层式逻辑寻址方式确定 LonTalk 数据包的源节点和目的节点和在路由器中实现不同信道间的数据传输。分层式逻辑寻址如图 2-13 所示，分层式逻辑寻址可以进行寻址整个区以及子网、节点的搜索。

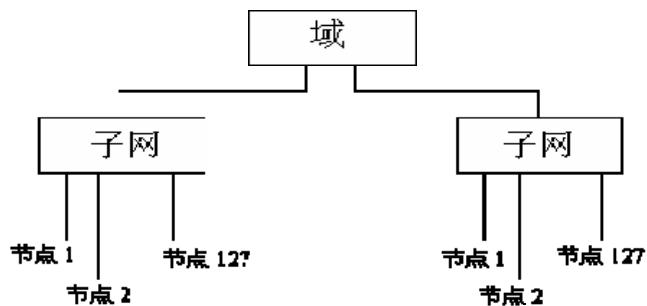


图 2-14 为分层寻址示意图

(4) LonTalk 协议的 MAC 子层

不同的网络中采用多种介质访问控权的协议。LonTalk 协议使用带预测的 P 坚

持 CSMA(predictive-persistentCSMA)协议^[12]，在协议中采用智能节点根据网络积压参数等待随机时间片来访问介质，网络上的每个智能节点通过随机插入 O-W 个很小的随机时间片来进行调整与数据传送，O-W 个很小的随机时间片在避免了网络数据频繁碰撞。

(5)LonTalk 协议的链路层^[13]

LonTalk 协议的链路层提供错误检测的能力，但不提供恢复能力。同时根据底层物理介质模式不同采用不同的链路编码模式。在直接互连模式下采用曼切斯特编码等。

(6)LonTalk 协议的网络层

LonTalk 协议网络层主要提供网络管理的网络地址分配、出错处理、网络认证、流量控制、路由器的机制也是在这一层实现的。

(7)LonTalk 协议的传输层和会话层

LonTalk 协议的核心部分是传输层和会话层，传输层管理着报文执行的顺序、报文的二次检测。会话层主要提供请求和响应机制，远程数据服务就是在会话层实现。

(8)LonTalk 协议的表示层和应用层

LonTalk 协议的表示层和应用层提供以下几类服务：

A：网络变量的服务。当定义为输出的网络变量改变时，能自动把网络变量的值向下发送，使所有把该变量定义为输入的节点收到该网络变量的改变。

B：显示报文的服务。把报文的目的地址、报文服务方式、数据长度和数据下传并发送。当收到信息后，显示报文。

C：网络管理的服务。能够分配所有节点的地址，进行节点查询、节点测试和设置路由器的配置表。

(9)LonTalk 协议的报文服务

LonTalk 协议提供了四种类型的报文服务，除了请求、响应是会话层实现的，其他三种都在传输层实现^[14]。

A.应答服务

网络中的智能节点通过报文发送与接收实现数据通信，在应答服务中发送方若没有收到接收方的全部应答报文则可以重新发送全部报文，直到全部正常为止。网络处理器专门负责报文应答服务数据自动处理。

B.请求、响应

由应用处理器实现请求与相应报文数据处理，适用于远程调用和客户、服务

器交换环境。

C.重发服务

应答报文丢失或者异常，发送方会重新发送该批次数据，该种方式主要用于智能点较多的广播，重发服务在一定程度上缓解了网络阻塞。

D.非应答方式

节点间不需要考虑应答和重新发送异常应答报文以及可靠性要求不高但速度较快、只发送一次即可环境下。

2.3.2 .Net 技术

2.3.2.1 .NET 与 Web Services

(1) .NET 思想及核心组件构成

自从 J2EE 推向市场以来，以其安全性、稳定性和扩展性得到了开发者的青睐，迅速成为大型 Web 应用的首选标准。作为应对，微软在 2000 年推出了.NET^[14]技术。

.NET 包括平台和知识两部分。

.NET 平台包括开发工具、服务器、.NET 积木块服务（BBSs）和设备软件。其中，开发工具包括.NET Framework 和 Visual Studio .NET。.NET Framework 和 J2EE 相对应，包括.NET CLR（公共语言运行环境）和公共类库。Visual Studio .NET 是 Visual Studio 开发工具的.NET 版本，包括 VB、VC、VBScript、JScript 等的新版本，以及一门新的语言 C#，并基于.NET Framework 开发。

服务器包括.NET Enterprise Servers 和.NET Server。其中.NET Enterprise Servers 包括 SQL Server、ISA Server、Biztalk Server 等系统服务平台^[15]。

.NET 积木块服务（.NET Building Block Services）是指基于.NET 技术构建的基于 XML 的基础性的 Web 服务，如文件存储、用户性能管理、日历管理等。这些基础性的 Web 服务将极大地扩展.NET 服务的功能性。

设备软件是指应用在 Smart Devices 上的软件，使这些设备可以象 PC 一样享受.NET 的强大功能。

.NET 知识包括.NET 开发思想、开发方法和用户开发经验，主要体现在 MSDN 中。

(2) Web Services

Web Services 是.NET 的基础技术之一。Web 和 Internet 技术的出现，促进了企

业业务模式的变革。从 B2C 到 B2B，直到 E2E（Enterprise to Enterprise），关键在于企业应用集成（EAI）技术的发展与推广。

实现 EAI 最理想的解决方法是标准化一个企业中的体系结构和框架，但这往往不现实，因为不可避免需要采用不同公司的软件包。因此，需要有这样一个机制，使得内部和外部的系统可以用一种松散耦合的模式集成。解决方法应是：将接口和实现相分离，公共的接口不变，实现变了以后，只要维护接口和实现之间的映射即可。

Web services 的出现，正是为了解决这些问题。可以将 Web services 想象为附加在现有应用或资源上的一个“包装”，通过这个包装，可以将应用或别的资源封装为一个服务。而 Web services 能确保这个服务有与技术无关的、通用的接口描述，并有统一的方式进行登记，便于访问和使用^[16~17]。

（3）.NET Framework

1) .NET Framework 是.NET 应用开发的基础平台。其由两个主要组件组成：公共语言运行时（CLR）和.NET Framework 类库^[16]。

CLR 是一个软件引擎，即加载和执行应用程序，类似于 Java 虚拟机（JVM）的作用。与传统的 Windows 应用程序引擎相比，CLR 提供了许多新的特性：应用程序的代码的执行受到控制。这样可以更方便地提高应用程序的执行效率，也提高了安全性，避免应用程序执行非受控操作。.NET 中有一个概念：部件（Assembly），一个.NET 应用程序是通过部件的形式进行发布的。部件内的代码是 MSIL（微软中间语言）代码，而在 CLR 将应用程序加载后才将 MSIL 代码翻译为适合本机的机器代码。由于有对从 MSIL 代码到本机代码翻译的控制，使得 CLR 可以管理应用程序的执行并防止各类问题的发生。这也就有了术语已管理代码（Managed Code）。

应用程序能方便地安装、升级，在同一主机上可以方便地运行应用程序的不同版本。部件中除了 MSIL 代码之外，还有两个部分，就是元数据（Metadata）和清单（Manifest）。其中元数据描述 MSIL 代码正确执行所需的各种相关数据类型，而清单则列出了部件中所有的文件和软件组件，并指出 CLR 在哪里可以找到具有应用程序运行所需的组件的其它部件。有了这两部分之后，部件就可以实现“自描述”，并彻底解决了所谓的“DLL Hell”，使得在将应用程序 XCOPY 至其它主机后能直接运行。

应用程序的编辑延迟至真正需要执行其的时候，避免编译那些不常用的代码。CLR 提供了 JIT（Just-in-Time）编译功能，实现代码的及时编译。当然，这样可能会在

某主机初次运行应用程序时有额外的开销，但及时编译从总体上提高了应用程序执行的效率，并充分利用了机器的性能。

减少因编程错误带来的 Bug 和安全漏洞的数量。CLR 通过监控代码以确保其不会有最常见的编程错误，例如试图使用一个整数作为函数指针，或者在载入数据时覆盖代码（由于缓存溢出）。这样也可以使恶意代码的运行更为困难。

提供应用程序执行所需的底层操作管理。CLR 中有 Garbage Collector、Security Engine、Debug Engine、Exception Manager 和 Thread Support 等功能组件，分别提供应用程序执行所需的垃圾清理、安全控制、调试控制、例外管理和线程支持等功能。

在已管理代码和未管理代码间起中介作用。CLR 提供了 COM Maeshaler (COM 整合器) 功能组件，使得.NET 代码可以与现存的 Windows 库和 COM 组件结合起来，并将一个应用程序逐渐地从老平台迁移到新平台上来。

2) .NET Framework 的类库和公共语言规格

.NET Framework 提供了一套 OS 级的对象函数库，可供支持.NET 的所有程序语言调用。它将原有的 Windows APIs、MFC 等的功能集于一身。同时，.NET Framework 提供的类库还按树状结构组织其命名空间 (Namespace)，利于程序员查找和使用。类库树状结构如图 2-15 所示^[17]。

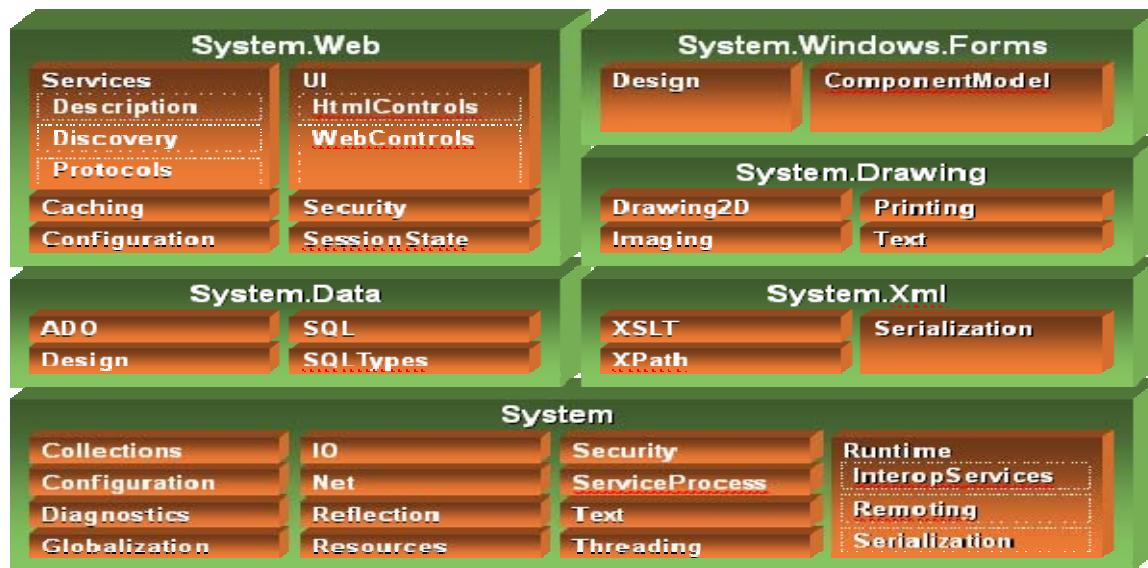


图 2-15 类库树状结构架构示意图

System 是.NET 对象结构中最重要的对象群集。在.NET 对象结构中，所有对象都是由 System 中的 Object 继承出来的。Object 声明了六个成员函数：Equals、

`GetHashCode`、`GetType`、`ToString`、`Finalize` 和 `MemberwiseClone`，即所有.NET 对象都拥有这六个成员函数。

`System.Data` 包含了存取数据库所需的各类对象和命名空间。.NET 使用 ADO.NET 取代原有的 ADO 结构^[18]。

`System.XML` 提供了.NET 应用程序处理 XML 文件的能力。`System.XML` 提供了符合 XML 1.0、DTD、XML Schema、XSLT、DOM 以及 SOAP 等标准的各类对象。

`System.Drawing` 封装的是用来存取 GDI 系统的相关对象。

`System.Windows.Forms`: 这是.NET 开发 Windows Forms 的基础，提供了许多 Windows Forms 组件和控件。

`System.Web`: 使程序具有控制 Web Server 的能力。它提供的对象模型可以让程序存取 Client 的请求信息、Cookie 等，并能控制 Web Server 的返回信息。其两个子命名空间 `System.Web.UI` 和 `System.Web.Services` 是.NET Web 应用开发的基础。`System.Web.UI`: 是.NET 中 Web Form 的基础。提供了 Web 界面所需的控件和相关的控制功能。

`System.Web.Services`: 提供 XML Web Services 的支持。

2.3.2.2 ADO.NET 数据库访问技术

(1) ADO.NET 设计目标

ADO.NET 是.NET Framework 提供的数据访问的类库，ADO.NET 对 Microsoft SQL Server、Oracle 和 XML 等数据源提供一致的访问。应用程序可以使用 ADO.NET 连接到这些数据源，并检索和更新所包含的数据。

(2) ADO.NET 组件及运行机制

ADO.net 采用多层架构，提供机制使应用程序能处理不同的数据源，不像以前的数据技术，需要由开发人员根据不同环境来定义一系列过程才能处理不同数据源。数据提供程序是 ADO.net 中的类的集合，它为应用程序和数据源之间建立了一座桥梁。数据提供程序主要组成部分如图 2-16 所示^[18]:

`ConnectionOp` 使用该对象建立与数据源的连接。

`Command` 使用该对象执行 SQL 命令和存储过程。

`DataReader` 提供了一些从查询提取的数据中执行快速只读、向前访问的机制。

DataAdapter 该对象执行两个任务。一个是由它负责把来自数据源提取的数据填充 **DataSet** (它是一个脱机模式下的表和关系集合); 另一个任务是使用它对数据源进行修改。

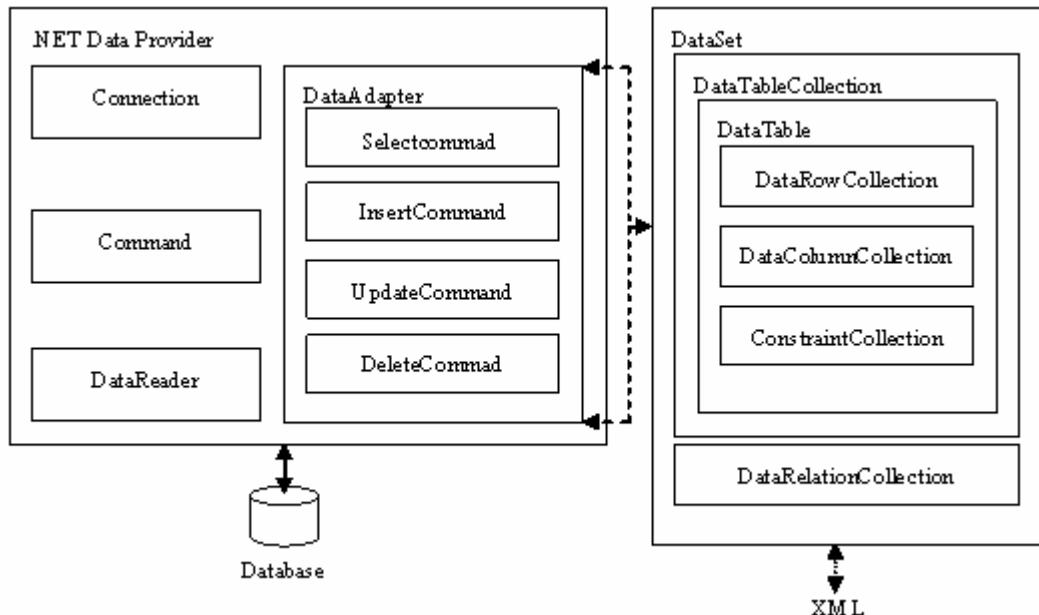


图 2-16 ADO.NET 的对象结构简图

ADO.NET 并没有包含一个通用的数据源提供程序对象。它为不同类型的数据提供了特定的数据提供程序，每一个程序对 ConnectionOp, Command, DataAdapter, DataAdapter 提供了最优化类。数据提供程序中提供了 4 类^[19]:

- (1) SQL Server 数据提供程序，提供了对 SQL Server 数据库的高效访问能力。
- (2) OLE DB 数据提供程序，提供了对具有 OLE DB 驱动程序的任何数据源的访问能力。
- (3) Oracle 数据提供程序，提供了对 Oracle 数据库的高效访问能力。
- (4) ODBC 数据提供程序，提供了对具有 ODBC 驱动程序的任何数据源的访问能力。

在这四类中，开发人员选择数据提供程序时，首先要使用特定数据源定义的数据提供程序。若没有，则使用 OLE DB，此类主要是通过数据提供程序，对于主流数据源都支持。最后选用 ODBC 驱动程序。

2.3.3 B/S 与 C/S 结构

2.3.3.1 C/S 结构

C/S 结构，两层架构，即客户机和服务器(Client/Server)结构，此架构充分利用了两边硬件环境的优势,将任务合理的分解给了客户端和服务端，降低通信开销。客户端有一个或者几个用户同时运行客户端程序，服务端也有两种情况：一种是数据库服务端，通过客户端数据库连接到服务端；另外一种是服务端是 Socket 服务端，服务端通过 Socket 与客户端通信。服务端一般采用高性能小型机，并采用大型数据库系统做支撑^[20]。而客户端则需要根据不同操作系统采用不同版本的客户端软件。在 C/S 架构，客户端要分担服务端部分功能，例如：由客户端实现业务逻辑和界面展示，通过与数据库交互实现客户端所需要的功能。

C/S 架构的优缺点：

- ① 客户端界面和操作可自定义。
- ② 安全性能高。
- ③ 响应速度快。
- ④ 缺点：
 - ⑤ 只适用于局域网。
 - ⑥ 需要安装客户端，使用群体受到限制。
 - ⑦ 维护成本高。服务端升级后，需要所有客户端一起升级。

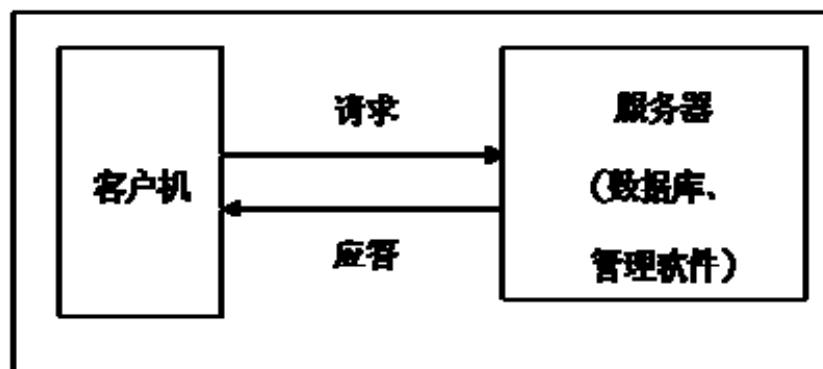


图 2-17 C/S 结构

在 C/S 结构图中：客户机 (PC 机) 向服务器 (数据管理 Server) 发出请求信息，服务端在将数据处理结果返回给客户端。该架构提供一个很好的安全保护和完整的数据处理，可以多个用户同时访问同一个数据库。

2.3.3.2 B/S 结构

B/S 结构，即浏览器/服务器(Browser/Server)结构，主要事物逻辑在服务端实现，客户端只负责页面展示，只需要普通浏览器就可以，其相关业务逻辑在 WebApp 服务端和数据库 DB 来实现，这样就减轻了客户端压力。

在 3 层结构系统中，将系统分为：“客户层”(用户界面)，“应用层”(商用逻辑)，“数据层”(数据库)。客户端只负责和应用层处理结果的展示。应用层软件部署在服务端，与客户端分离，当应用端应用发生变更时，不影响客户层和数据层。

B/S 优缺点^[21]:

- ①客户端不需要安装，通过 Web 浏览器即可。
- ②适用于广域网，通过一定权限控制实现多用户访问，交互性高。
- ③应用层，数据层升级不影响客户层。
- ④缺点：
 - ⑤跨浏览器时，有时候存在展示不尽人意。
 - ⑥要实现类似 CS 程序的程度需要花费很大精力。
 - ⑦速度和安全性上的设计成本比较高。
 - ⑧由于使用请求-响应处理模式，需要经常刷新页面。

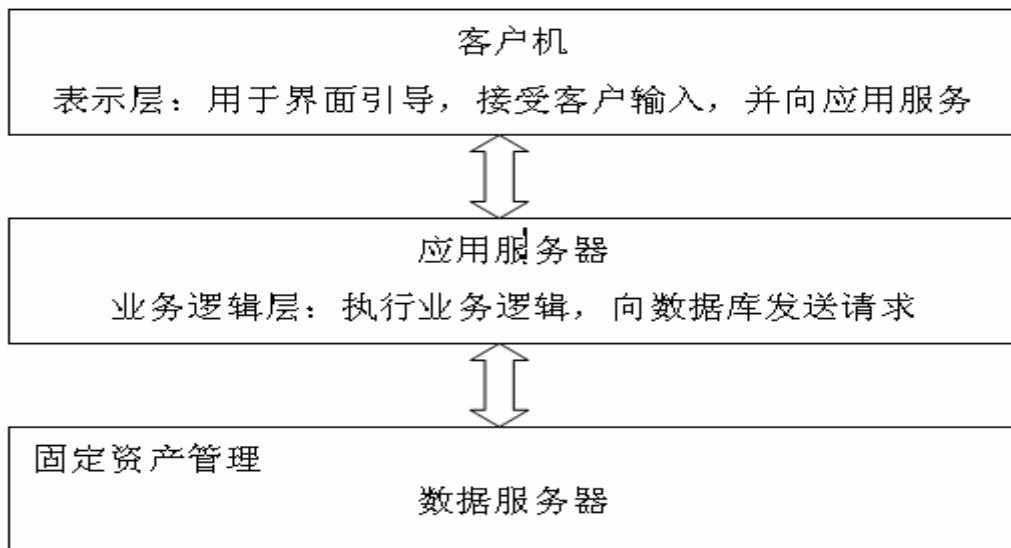


图 2-18 B/S 结构

在上图 B/S 结构中：中间件应用服务器实现客户机和服务器间的连接和通讯；实现与数据库间的高效连接；这三层相互独立，改变任何一层都不会对另一层产生影响。

2.3.3 C/S 和 B/S 比较

图 2-4 C/S 和 B/S 其它对比

| 对比项 | C/S 架构 | B/S 架构 |
|---------------|---|---|
| 技术成熟度 | C/S 架构简单，相对成熟，难出现新技术 | Web Service、Web Form 等新技术不断出现 |
| Internet 的支持度 | 受制于数据库类型 | 针对 Internet 应用而设计，只需 HTTP 协议支持 |
| 人机交互性 | 客户端程序采用标准 Win32 应用。普通的程序员就可以设计复杂交互程序。 | 交互能力差 |
| 维护工作量 | 需安装客户端软件，维护工作量大 | 客户端一般不需要维护 |
| 和数据库连接性 | 客户端直接连接数据库 | 客户端不需要连接数据库 |
| 宽带消耗程度 | 客户端和服务端主要是指令集和结果集的交互，带宽要求低。 | Web 发出的数据，除了结果数据外，还需要页面本身 |
| 数据安全 | 主要在局域网内使用，数据在传输安全通常由数据库厂商来保证，应 | 由于在 Internet 上采用明文形式传输，为保证安全，一般采用 SSL，VPN 等技术 |
| 运行性能 | 用程序不需要考虑传输安全问题。中小规模应用软件的运行性能差别不大，但对于大型企业级应用，B/S 架构容易产生各种性能瓶颈。 | |

以上可以看出 B/S 体现方便的特征，但由于客户端维护成本高。而 C/S 架构只需要管理服务端，所有客户端只需要浏览器，无需要任何维护成本，无论客户端有多少用户，或者增加多少客户端，都不会增加其维护成本。这将是信息化发展的主要方向。

2.4 本章小结

本章主要完成了楼宇自动控制系统的需求分析与开发工具介绍。介绍了楼宇自动化需求以及系统设计目标，分别对系统业务需求、功能需求分析进行分析，然后提出可执行性解决方案。接着最后一节介绍了开发楼宇自动化系统的 LonWorks 核心技术，内容包括：LonWorks 简介、硬件组成、LonTalk 通信协议，网络结构和 .Net, Web services 技术。

第三章 系统详细设计

3.1 系统结构详细设计

3.1.1 Web 结构设计

系统总体运行流程如图 3-1 所示，本系统数据库采用 SQL Server2005 大型关系型数据库，WEB Server 采用 MS IIS V7.0。

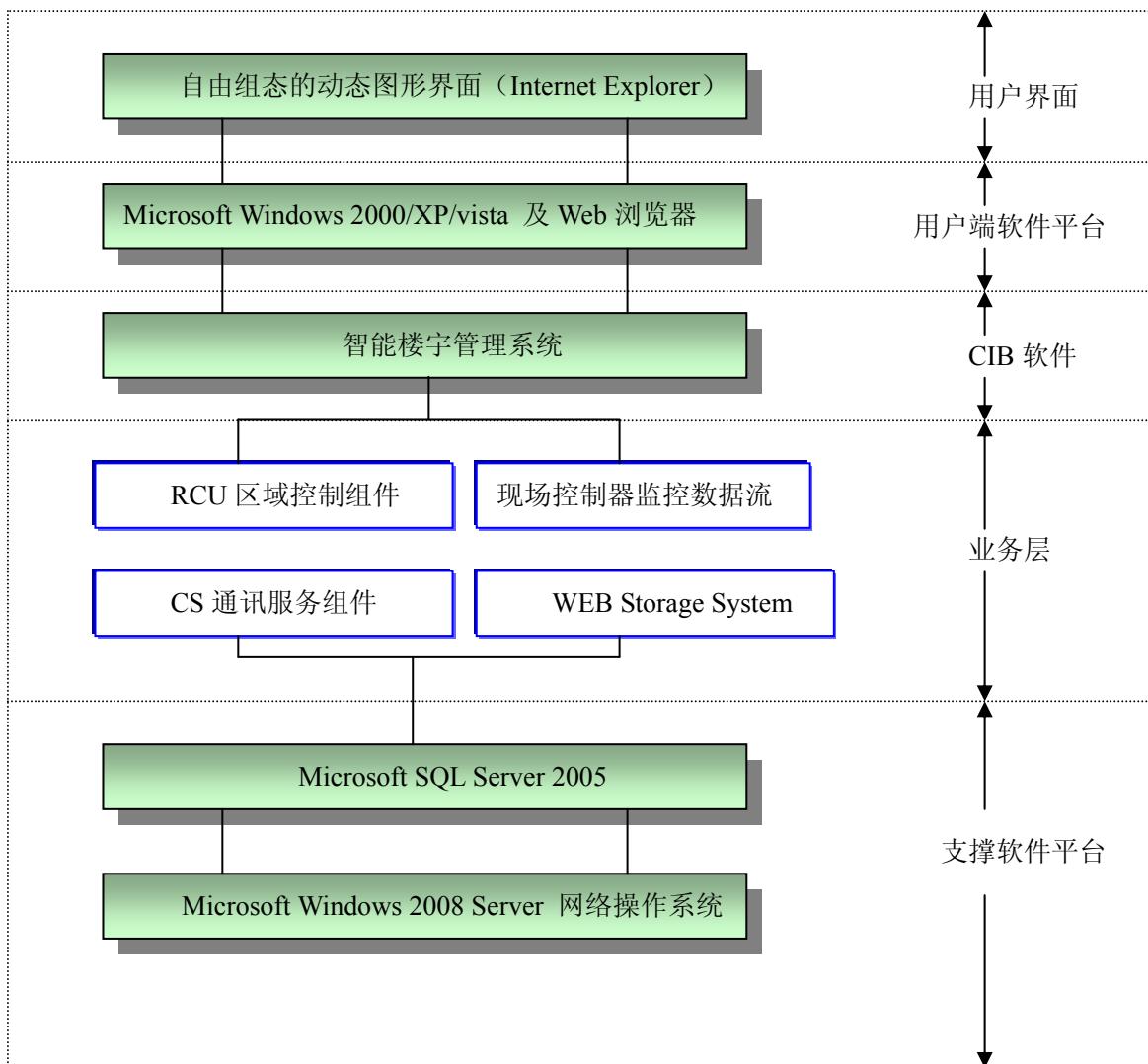


图 3-1 系统总体工作流程

3.1.1.1 Web 服务层设计

系统应用服务是由“SCT 系统配置工具”及“PCT WEB 页面生成工具”两部分软件功能模块组成。

(1) SCT 系统配置工具基本结构

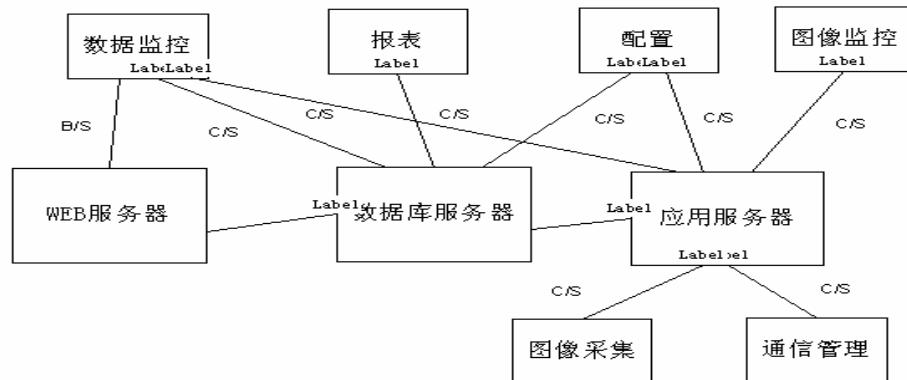


图 3-2 SCT 系统配置工具基本结构

(2) PCT WEB 页面生成工具工作流程

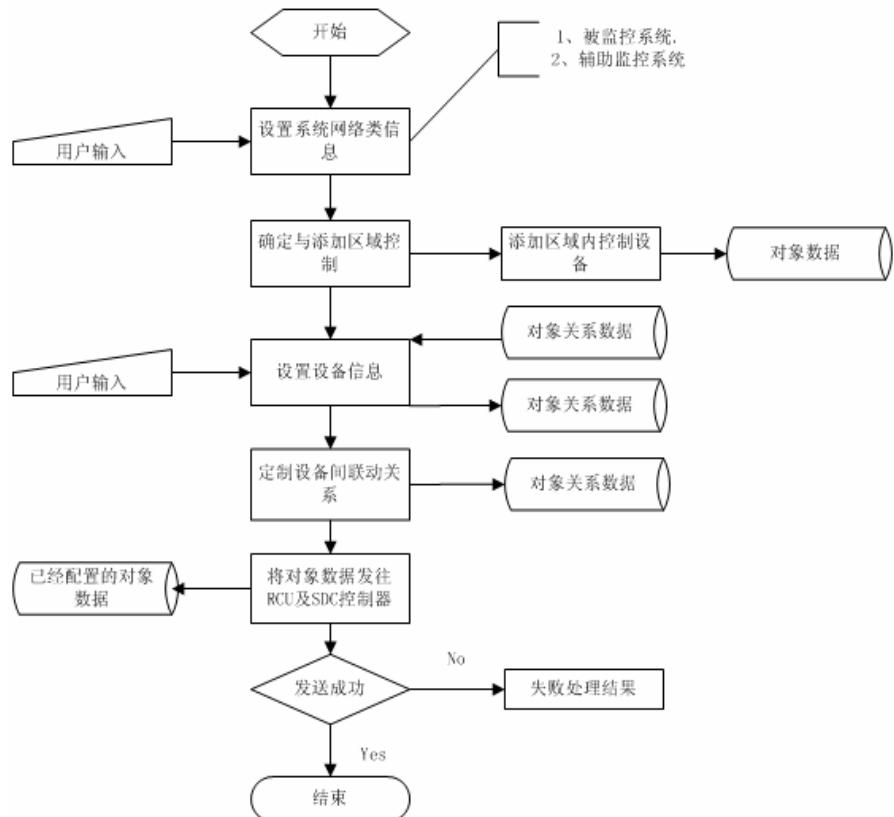


图 3-3 PCT WEB 页面生成工具工作流程

PCT 使用“物件导向设计”和 WEB 页面嵌入技术。使用者无须特别的软件编程技能，也无须撰写程式，只要将建筑物的平面图载入系统，然后在萤幕上使用鼠标“拖”、“放”功能模块，就可在最短时间内建构出一套专业的自动化系统。

3.1.1.2 Web 数据处理层设计

3.1.1.2.1 CS 通讯服务基本处理流程

CS 通讯服务模块是整个系统正常运作的枢纽，是系统 B/S、C/S 运行模式的“S”端。负责将底层控制器的运行数据写入系统数据库，并反馈执行动作的指令，同时将底层数据的变化通过 WEB 组态图形界面显示，接收从 IE 浏览端反馈的用户操作指令，控制系统按用户的指令运行。

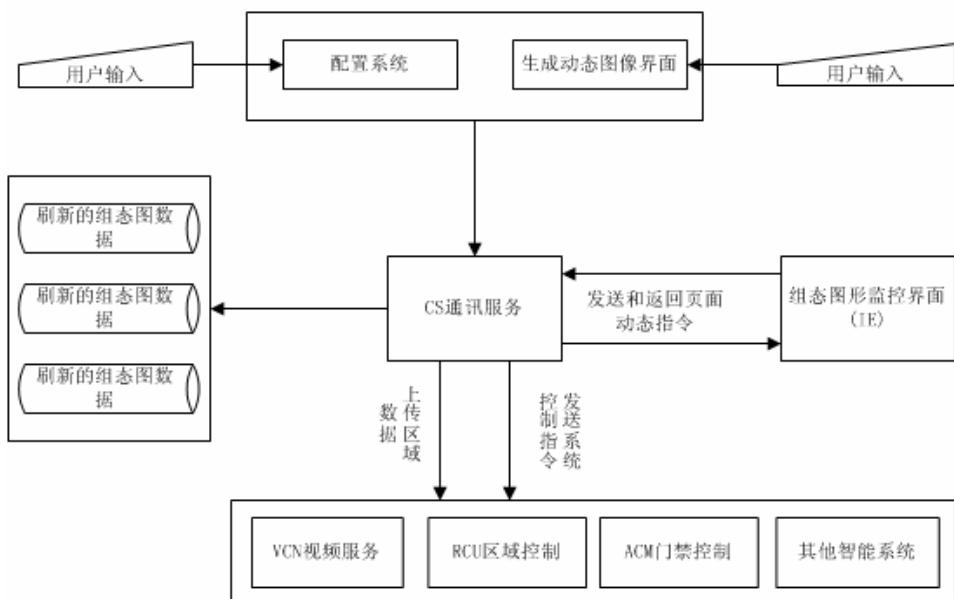


图 3-4 CS 通讯服务基本处理流程

3.1.1.2.2 RCU 基本处理流程

RCU 区域控制器负责对 SDC 智能控制器的控制。以下将从“对 SDC 的在线配置”、“实测数据的处理”、“系统控制指令的执行”、“告警信息的处理”四个方面对 RCU 的工作流程进行说明。

(1) 系统对 SDC 控制器配置的处理流程：

用户在 SCT 系统配置工具中完成整个系统配置后，运行对 SDC 控制器的在线配置指令，通过 RCU 控制即可完成对多个不同地点的 SDC 控制器上不同监控点属性的自动配置。

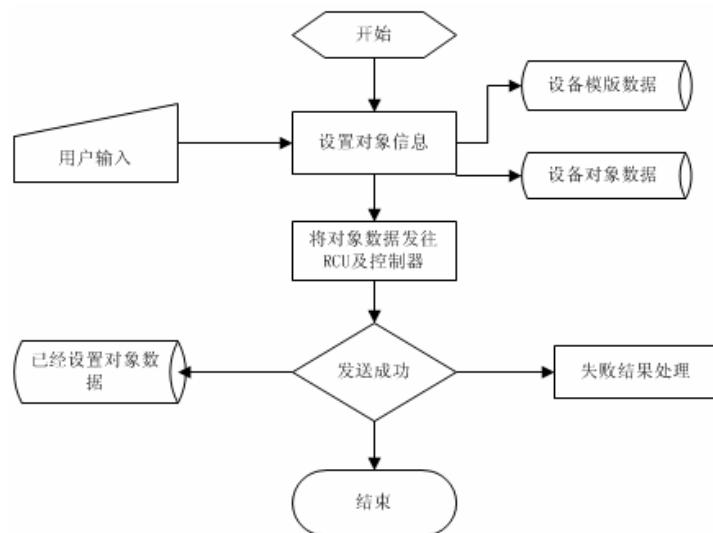


图 3-5 系统对 SDC 控制器配置的处理流程

(2) 告警数据处理流程

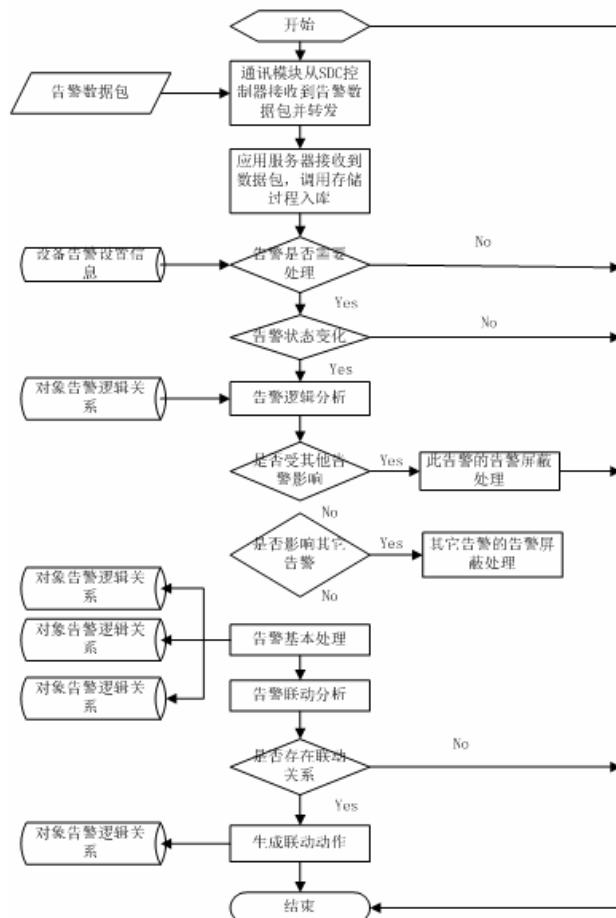


图 3-6 告警数据处理流程

3.1.1.2.3 用户监控端基本处理流程

(1) 用户处理过程

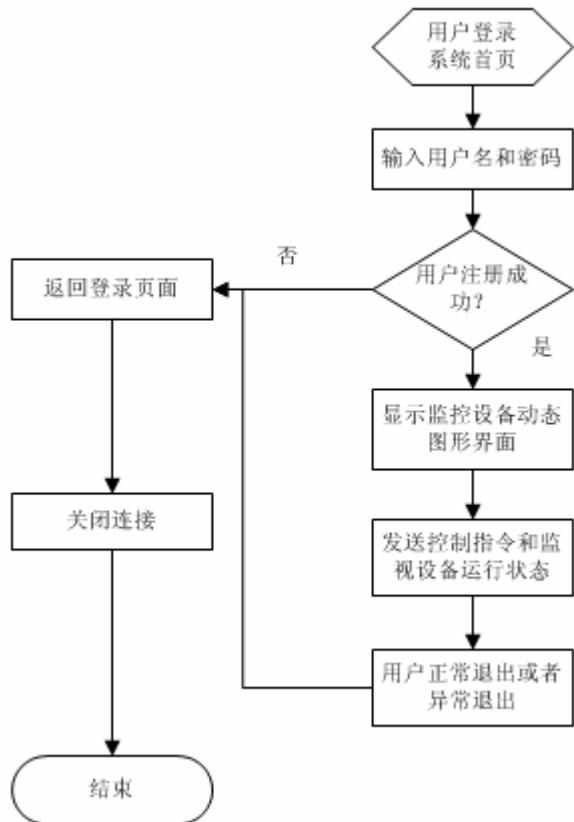


图 3-7 用户处理过程

(2) 用户操作权限管理过程

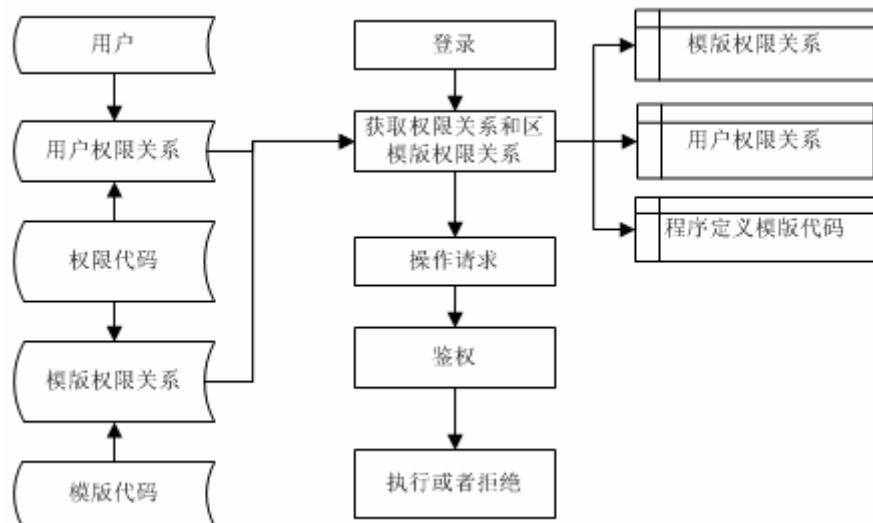


图 3-8 用户操作权限管理过程

通过这样的结构，授权的用户可以在远程获得建筑设备每一相关数据，除了数据监测和报警功能之外，还有比如数据记录、趋向预测、基本维护等功能。

3.1.2 基层结构设计

3.1.2.1 Lon 网络集成

LON 网络的各节点的组网与网络变量的绑定都是通过 Echelon 提供的 LonMaker for Windows 集成工具来实现的，前面已经对 LonMaker for Windows 工具做了简单的介绍

本节介绍如何通过 LonMaker 进行节点组网。

具体可分为以下步骤^[26]：

(1)通过 LonMaker 程序建立用于维护整个现场总线网络数据库名，作为 LNS DDE Server，管理网络变量

(2)指定网络接口卡，采用 PCLTA-20 网络接口卡实现插即用型功能。

(3)选择网络管理模式，在线网络模式情况下网络中节点配置变化会自动传输到下层网络，而离线模式前提下，只是本地立即生效。

(4)向网络中添加节点，如添加的是 LonPoint 节点则可以从已有的 LonPoint 节点图形中选择，若所添加的节点是自行设计的节点，可从基本图形中选择 device 图形。

(5)添加节点的功能块与网络变量之间的绑定。节点网络变量之间的绑定必须要求所绑定的网络变量都是相同的类型，必须是输出网络变量和输入网络变量之间的绑定。

(6)配置节点（本设计中主要是针对 DO,AI 两节点）

对于 AI 节点，设计要实现的功能是根据测量的温度感器的电阻值，输出相应的温度，AI 节点配置就是在配置 AI 的工作模式为测量电阻模式，然后建立起测量的电阻值与输出温度的对照表。最后需要将这些配置信息装载到 AI 节点。

配置工作模式和温度对照表窗口如下：



图 3-9 工作模式配置窗口

This screenshot shows the 'Translation' configuration window for AI-1. It displays a table of measured values versus output values. The table has two columns: 'Measured Value' and 'Output Value'. A third column 'Measured Value' and 'Output Value' is also present. Annotations point to the 'Download conversion table to node' button (labeled '#1e') and the 'Save' button.

| Measured Value | Output Value | Measured Value | Output Value | Download conversion table to node |
|----------------|--------------|----------------|--------------|-------------------------------------|
| 0 | 40.999 | 10 | 6650 | #1e |
| 1 | 31.111 | 11 | 6790 | <input type="button" value="Load"/> |
| 2 | 22.222 | 12 | 61240 | <input type="button" value="Save"/> |
| 3 | 14.333 | 13 | 24560 | |
| 4 | 9.556 | 14 | 27310 | |
| 5 | 6.375 | 15 | 31080 | |
| 6 | 4.444 | 16 | 32953.4 | |
| 7 | 3.111 | 17 | 0 | |
| 8 | 2.222 | 18 | 0 | |
| 9 | 1.444 | 19 | 0 | |

图 3-10 电阻温度对照表配置窗口

由于对 DO 节点的使用比较简单基本不需要配置, 所以对 DO 节点的配置略去。

3.1.2.2 现场网络与上位机通信实现技术

现场网络与上位机的通信需要用到 LonWorks DDE 技术。LonWorks 技术中的 DDE(Dynamic Data Exchange)技术提供了与 Windows 应用程序之间信息共享的软件支持。LonWorks 技术提供 LonManager DDE 和 LNS DDE Server 两种软件工具支持应用程序间的 DDE 方式, 通过 DDE 技术我们就可以实现对网络变量的监测和控制^[27]。

在设计中采用 LNS DDE Server2.0 作为 DDE 服务的提供者。前边已经对 LNS DDE Server2.0 做了介绍这里不再赘述。

DDE 提供的服务类型有：开始、结束、请求、建议，分别表示开始一个对话、结束一个对话、Client 对源服务器中某种特定信息的在线请求和 Client 对源服务器中某特定信息持续修正的请求。利用 DDE 对话请求选项，可以实现 Hot 方式（无论何时 Item 项改变，服务器自动发送给目标 Client 新值）和 Poke 方式（当 Client 发出请求时刷新 Item 值）。

采用 LNS DDE Server 服务器，Windows 应用程序可以监视任一网络变量或者修改任一网络变量的值。而所有这些的实现并不需要编程。

Visual Basic 对 DDE 技术的支持，在 VB 的界面设计中，常会用到一些诸如文本框、标签框等控件，事实上 VB 已经为这些控件提供了与 DDE 之间的接口。在 VB 中与 DDE 通信只需要设置好三个属性就行了。它们分别是 LinkTopic 属性、LinkItem 属性和 LinkMode 属性。其中 LinkTopic 属性设置了被请求服务器的名字和谈话的主题；LinkItem 属性指定了对话的项，也就是网络变量的名称及其所在节点；LinkMode 属性指定对话的模式，一般选择 Automatic 模式。例如，在设计中需要监视底层上传的代表温度的网络变量值则各属性设置如下（用一个 label 控件来显示）该控件的三个属性设置如下^[28]：

```
LinkTopic = "LNSDDE|www.Subsystem 1.LMVN"  
LinkItem  = "AI- 1.AI- 1.Analog"  
LinkMode = 1-Automatic
```

当要改变网络变量的值时需要用到 LinkPoke 方法在设计中需要通过改变网络变量的值来控制电饭煲继电器开关的开合。方法如下^[29]：

将一个名为 relay 的 label 控件的属性设置如下：

```
LinkItem = "fuzzy.NVI1"  
LinkTopic = "LNSDDE|www.Subsystem 1.DevNV"  
LinkMode = 1-Automatic
```

在程序中可以通过下边代码来更改 fuzzy 节点的输入网络变量 NVI1 的值

```
relay=1  
relay.LinkPoke
```

对于程序中其他的网络变量的监视和改变都是采用以上的方法这里不再赘述。

3.2 系统功能设计

3.2.1 系统总体结构设计

系统功能采用了分层结构，分层结构有利于系统的维护和更新，同时有利于后期二次软件开发与扩展。其总体结构如 3-15 所示

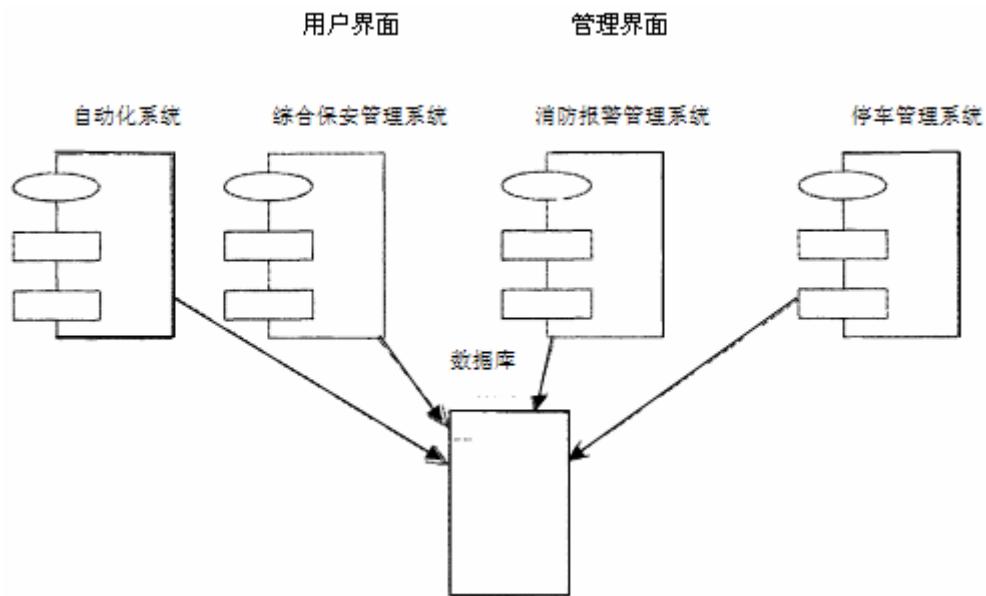


图 3-15 系统功能总体结构

3.2.2 停车管理子系统

(1) 停车管理系统框架

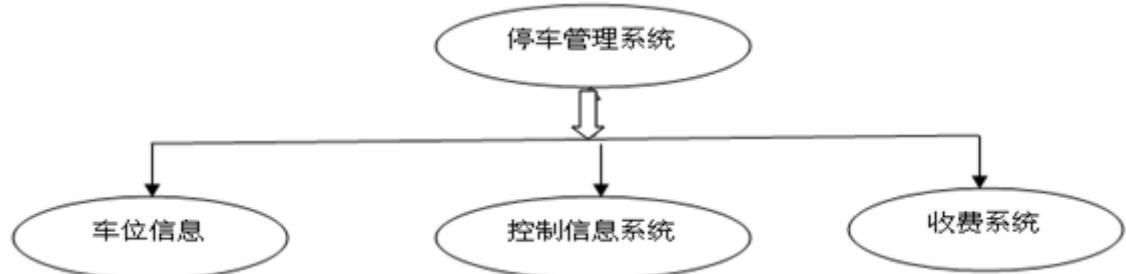


图 3-16 停车管理系統框架

该模块主要处理系统进出管理，以及车位信息空闲与否管理，以及停车收费管理等功能。其处理过程如 3-17 所示

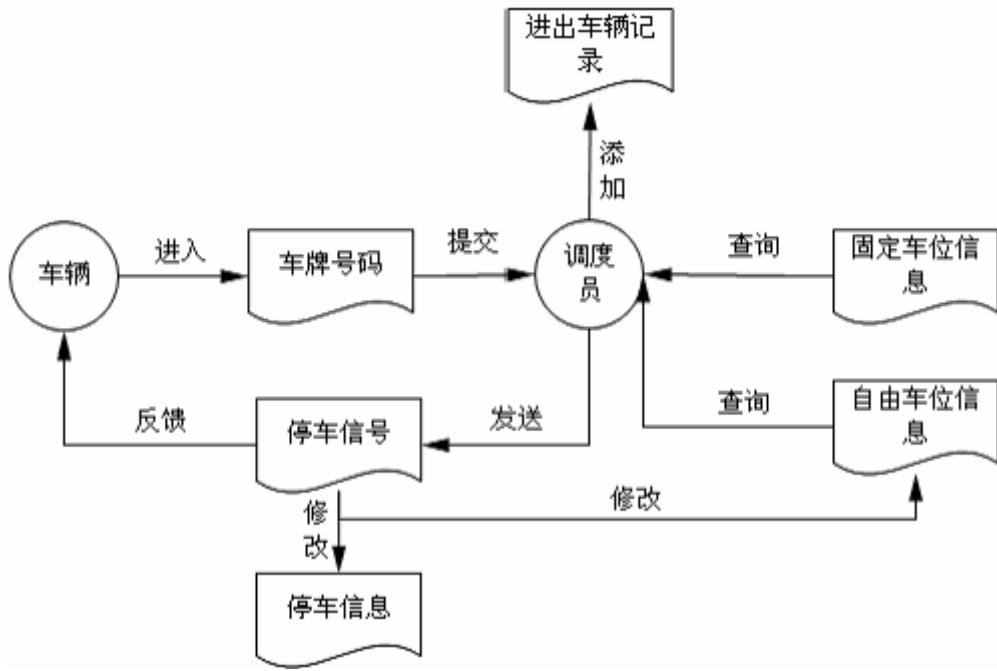


图 3-17 停车处理子系统大致处理过程

(2) 系统包含的对象类

系统中主要有三个累、每个类包含的主要属性和方法如表 3-1

表 3-1

| 核心库表及其主要字段 | | |
|---------------------|-------------------|------------------|
| CarControl | DeviceInfo | Charge |
| TotalNumber:integer | DeviceNo:Integer | Price:Currency |
| UsedNumber:integer | DeviceName:String | PlaceNo:Integer |
| EmptyNumber:Integer | | GoTime:Date |
| | | ParkFee:Currency |
| | | ComeTime:date |
| 对象类 | | |
| FullCar() | ComeAction() | ChargeForm() |
| AddCar() | GoAction() | Charging() |
| GoCard() | Exception() | |
| | | |

车位控制类中 TotalNumber 车的总量，UsedNumber 为已经被使用的位置量，EmptyNumber 剩余车位，FullCar()查询是否空闲车位，AddCar()表示可以再增加停车数量；GoCar()表示车位已经腾空。

设备信息类中的 DeviceNo 记录设备号，DeviceName 记录设备名。ComeAction 表示当有汽车进入时候出发相关位置信息被占用状态更新：GoAction 函数表示但

车离开车库时候出发位置信息空闲数据更新、Exception 表示异常情况的数据更新。

收费类中 PlaceNo 表示车位号；ComeTime 停车开始时间，GoTime 汽车离开车库时间；ParkFee 表示停车费用。

其函数代码片段如

```
Public class CarControl
{
    Public integer TotalNumber;
    Private Integer UsedNumber;
    Private Integer EmptyNumber;

    Public Carcontrol()
    {
    }

    Private Boolean FullCar()
    {
    }

    Private AddCar(Integer Number)
    {
    }

    Private Boolean GoCar(integer CarNumber)
    {
    }
}
```

3. 2. 3 自动化系统设计

自动化系统设计主要是楼宇中的空调系统、照明系统、机电系统排水系统等系统自动化控制，我们以空调系统为例，其控制用例为：

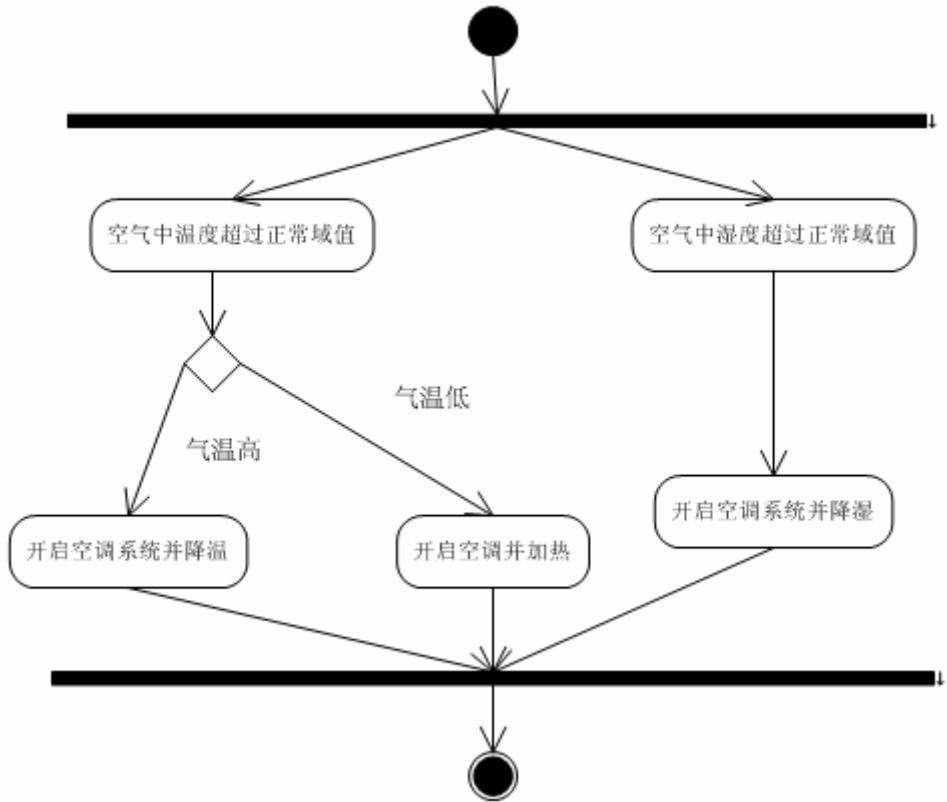


图 3-18 空调系统自动化控制状态图

空调系统是通过热敏、湿敏原件传感器来实现，但这些器件接收到外界刺激时就会触发温度调节的功能，通过调整电力、风力来使温度达到平衡点。而空调系统、机电系统、照明系统等楼宇设备间（包括楼宇各类传感器间）的数据传输、通信以及统一控制则在后面基于 Lon 现场中技术中进行详细介绍。

3.2.4 消防报警管理子系统设计

消防报警管理子系统主要是烟感探测器系统和温感探测器系统。系统中，当探测器发现有火情时能自动开启消防广播，播送火灾灾情，并启动警铃的鸣响。同时，系统将触发灭火设备的工作状态，自动灭火，并打开楼层排烟阀，缓解灾情，使楼层所有电梯迫降，自动切断相关电源，防止灾害进一步扩大。必要时，将消防报警管理系统与公安 110 联网，寻求 110 的帮助。

其报警用例如 3-19 所示

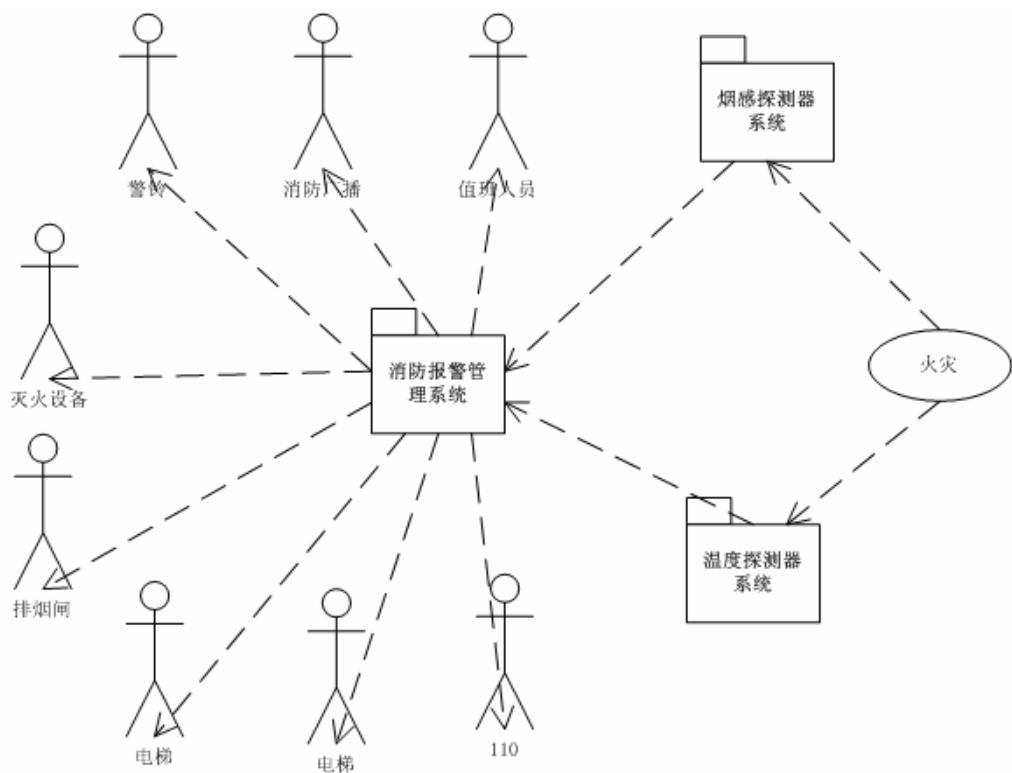


图 3-19 消防设计

在系统中我们可以看出，系统除了要具备探测器火情探测外，还应该联动发出火情告警、广播、以及自动拨打火警电话。通过系统中的联动模块来实现多个模块联动报警。我们可以通过如 3-19 结构模式建立联动报警。

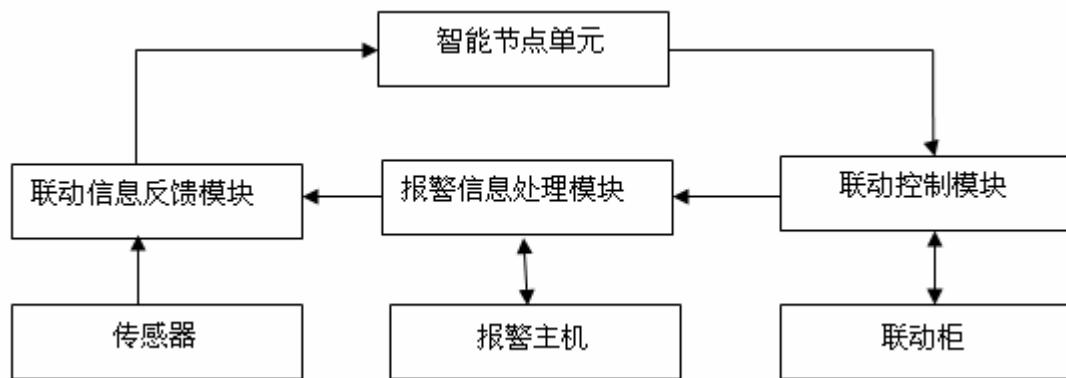


图 3-20 联动报警机制

3.2.5 综合保安管理系统的应用

综合保安管理系统的应用包括：防盗报警系统、闭路电视监控系统和电子巡更系统。在防盗报警系统在这里重点介绍一下门禁系统。

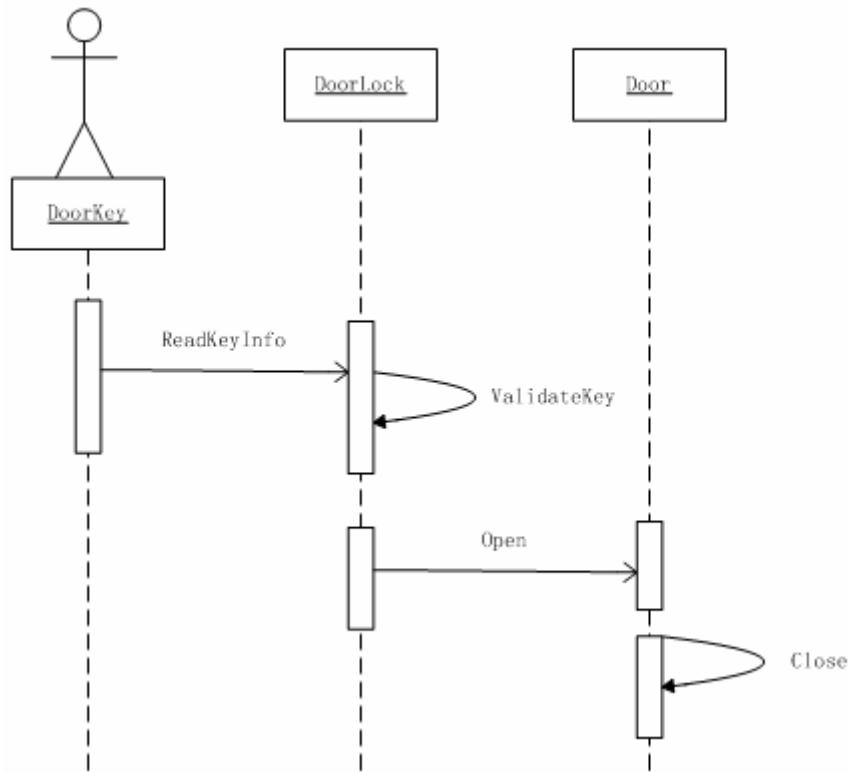


图 3-21 门禁系统

在门禁系统中（如 3-21 用例图）我们通过 DoorKey 上的用户信息，调用 DoorLock 对象中的 ReadKeyInfo()方法实现信息读取，然后通过 ValidateKey()函数进行信息稽核，若信息匹配正确则调用 Open()函数，然后开门后最后采取 Close()函数进行关门。

3.3 数据库设计

3.3.1 数据库逻辑结构(E-R)

我们主要以系统中部分数据库设计进行分析。系统的数据库设计是在对需要处理的数据表格的进行整理、分析的基础上进行的，设计结果如 3-22 图，其中对每一张数据表的名称、字段名称、意义、长度等作出定义，同时也对数据表之间的关系作出了说明。

(1) 逻辑结构设计 (E-R 图)



图 3-22 楼宇表 E-R 图



图 3-23 房间 E-R 图



图 3-24 周边设施 E-R 图

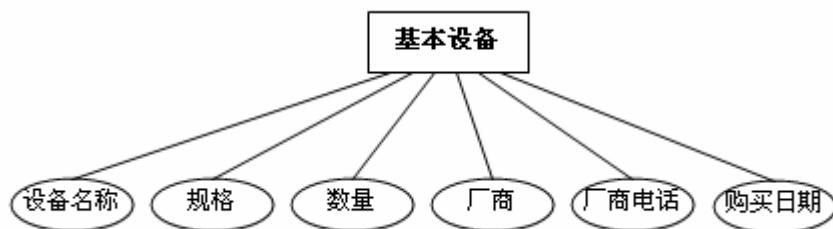


图 3-25 基本设备 E-R 图

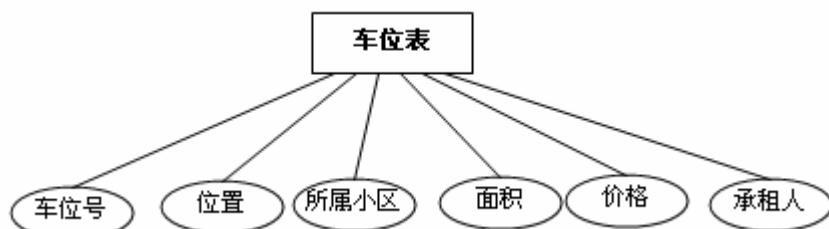


图 3-26 车位 E-R 图



图 3-27 管理 E-R 图



图 3-28 房间 E-R 图



图 3-29 房间 E-R 图

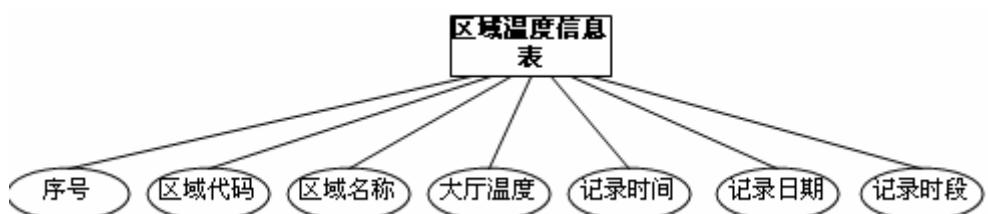


图 3-30 房间 E-R 图



图 3-31 房间 E-R 图

(2) 物理结构设计

1) 楼宇表(Pavilion)

| 列名 | 数据类型 | 是否为空 | 说明 |
|-----------|-------------|---------|--------------|
| PID | int | NOTNULL | 楼号, 设为主键、标识列 |
| CenShu | int | NOTNULL | 层数 |
| MianJi | Float | NOTNULL | 占地面积 |
| ChaoXiang | Varchar(10) | NOTNULL | 朝向 |
| AName | Varchar(50) | NOTNULL | 所属小区 |
| LeiXin | Varchar(10) | NOTNULL | 类别 |

2) 房间表(House)

| 列名 | 数据类型 | 是否为空 | 说明 |
|-----------|-------------|---------|--------------|
| HID | int | NOTNULL | 房号, 设为主键、标识列 |
| PID | int | NOTNULL | 所属楼号 |
| ZID | Int | NULL | 房主 |
| ChaoXiang | Varchar(10) | NOTNULL | 朝向 |
| MianJi | Float | NOTNULL | 面积 |
| FangXin | Varchar(10) | NOTNULL | 房型 |

3) 周边设施表(AreaK)

| 列名 | 数据类型 | 是否为空 | 说明 |
|---------|-------------|---------|---------|
| AKID | Varchar(50) | NOTNULL | 设施名称,主键 |
| UserID | Int | NULL | 负责人, |
| DianHua | Varchar(50) | NOTNULL | 联系电话, |
| LeiXin | Varchar(20) | NOTNULL | 类型 |

4) 基本设备表(SheBei)

| 列名 | 数据类型 | 是否为空 | 说明 |
|-------|-------------|---------|---------|
| SID | Varchar(50) | NOTNULL | 设备名称,主键 |
| GuiGe | Varchar(50) | NOTNULL | 规格 |

| | | | |
|------------|-------------|---------|-------|
| ShuLiang | int | NOTNULL | 数量 |
| ChangShang | Varchar(50) | NOTNULL | 厂商, |
| DianHua | Varchar(50) | NOTNULL | 厂商电话, |
| ShiJian | DateTime | NOTNULL | 购买日期 |

5) 车位表(Carbarn)

| 列名 | 数据类型 | 是否为空 | 说明 |
|------------|-------------|---------|------------|
| CID | int | NOTNULL | 车位号,主键、标识列 |
| WeiZhi | Varchar(50) | NOTNULL | 位置 |
| AName | Varchar(50) | NOTNULL | 所属小区 |
| MianJi | Float | NOTNULL | 面积 |
| JiaGe | Float | NOTNULL | 价格 |
| ShipToName | Varchar(50) | NULL | 承租人 |

6) 管理表

| 列名 | 数据类型 | 是否为空 | 说明 |
|--------|------|---------|----------------|
| UserID | int | NOTNULL | 员工号和业主编号一起设为主键 |
| ZID | int | NOTNULL | 位置 |

7) 管辖表

| 列名 | 数据类型 | 是否为空 | 说明 |
|--------|-------------|---------|----------------|
| UserID | int | NOTNULL | 用户号和小区名称一起设为主键 |
| AName | Varchar(50) | NOTNULL | 小区名称 |

8) 扶梯信息表

表名: liftinfo

| Name | Code | Data Type | Primary | Foreign key | Mandatory |
|------|------------|-------------|---------|-------------|-----------|
| 序号 | id | int | TRUE | FALSE | TRUE |
| 扶梯代码 | liftcode | varchar(4) | FALSE | FALSE | FALSE |
| 扶梯名称 | liftname | varchar(20) | FALSE | FALSE | FALSE |
| 扶梯状态 | liftstate | varchar(4) | FALSE | FALSE | FALSE |
| 记录时间 | recordtime | datetime | FALSE | FALSE | FALSE |

9) 灯光信息表

表名: lampinfo

| Name | Code | Data Type | Primary | Foreign key | Mandatory |
|------|------------|-------------|---------|-------------|-----------|
| 序号 | id | int | TRUE | FALSE | TRUE |
| 灯光代码 | lampcode | varchar(4) | FALSE | FALSE | FALSE |
| 灯光名称 | lampname | varchar(20) | FALSE | FALSE | FALSE |
| 扶梯状态 | lampstate | varchar(4) | FALSE | FALSE | FALSE |
| 记录时间 | recordtime | datetime | FALSE | FALSE | FALSE |

10) 区域温度信息表

表名: aircondinfo

| Name | Code | Data Type | Primary | Foreign key | Mandatory |
|------|------------|--------------|---------|-------------|-----------|
| 序号 | id | int | TRUE | FALSE | TRUE |
| 区域代码 | aircode | varchar(4) | TRUE | TRUE | TRUE |
| 区域名称 | airname | varchar(20) | FALSE | FALSE | FALSE |
| 大厅温度 | areastate | decimal(5.1) | FALSE | FALSE | FALSE |
| 记录时间 | recordtime | datetime | FALSE | FALSE | FALSE |
| 记录日期 | recordday | varchar(10) | FALSE | FALSE | FALSE |
| 记录时段 | recordhour | decimal(5.1) | FALSE | FALSE | FALSE |

11) 区域温度修正表

表名: temprevise

| Name | Code | Data Type | Primary | Foreign key | Mandatory |
|------|------------|-------------|---------|-------------|-----------|
| 区域代码 | liftcode | varchar(4) | TRUE | FALSE | TRUE |
| 区域名称 | airname | varchar(20) | FALSE | FALSE | FALSE |
| 修正温度 | revisedata | float | FALSE | FALSE | FALSE |

3. 3. 2 数据结构与程序的关系

下面将按照功能模块的顺序说明各大功能模块与数据表之间的管理:

(1) 接口程序与数据表

| 模块 | 功能模块 | 数据表 | 操作 |
|------------|----------|--------|----|
| 楼宇监控数据转换接口 | 自动扶梯数据转换 | 扶梯信息 | 写入 |
| | 区域温度数据转换 | 区域温度信息 | 写入 |
| | 楼宇灯光数据转换 | 灯光信息 | 写入 |

(2) 监控查询主程序与数据表

| 模块 | 功能模块 | 数据表 | 操作 |
|----------|------|--------|----|
| 自动扶梯信息查询 | 实时查询 | 扶梯信息 | 只读 |
| 区域温度信息查询 | 实时查询 | 区域温度信息 | 只读 |
| | 温度曲线 | | |
| | 温度修正 | 区域温度修正 | 读写 |
| 楼宇灯光信息查询 | 实时查询 | 灯光信息 | 只读 |

3.3.3 数据库具体设计

软件系统项目主文件，用于创建主程序，判断程序是否已经运行。如果程序已经运行，则不在创建主程序，确保只有一个程序运行。

软件系统主界面文件，使用 3 个 TDBGridEH 控件，用于显示读取的 excel 数据。1 个控件 TADOConnection，作为 3 个 TADODataset 控件的 Connection，每个 TADODataset 控件读取 excel 文件中的一个 sheet 并作为对应 TDataSource 控件的 DataSet。3 个 TDataSource 控件对应 3 个 TDBGridEH 控件作为他们的数据源。Ttimer 控件每隔一段时间执行一次 ontime 事件，把 TDBGridEH 中的数据写入对应数据库^[30]。在主界面生成时建立对 excel 文件的连接，并将文件位置写入注册表，方便以后变动时进行更改，不需更改程序本身。

数据库连接功能文件，定义了数据库连接 conn 的建立，断开，以及数据查询，更新，删除和数据格式转换等功能模块。具体函数如下：

```

function ConnectDB:boolean; //连接数据库
procedure Disconnect; //断开连接
procedure ExecSql(sSqlSen:String); overload; //执行单条更新、删除等 sql 语句
function QueryDB(sSqlSen:String):boolean; overload; //执行查询 sql 语句
function ExecSqlS(sSqlSens:TStrings):boolean; //执行一批更新、删除等 sql 语句

```

```
procedure ExecSql(sSqlSen:String;slPrmList:TStrings); overload;
function QueryDB(sSqlSen:String;slPrmList:TStrings):boolean;  overload;
procedure ExecSql(sSqlSen:String;oleVPrm:OleVariant); overload;
function QueryDB(sSqlSen:String;oleVPrm:OleVariant):boolean; overload;
procedure CloseQuery;//结束查询
```

对查询结果进行数据格式转换的函数：

```
function Field(nIndex:Integer):Variant;overload;
function Field(sFieldName:String):Variant;overload;
function FieldIsNull(nIndex:Integer):boolean;overload;
function FieldIsNull(sFieldName:String):boolean;overload;
function FieldAsString(nIndex:Integer):String;overload;
function FieldAsString(sFieldName:String):String;overload;
function FieldAsInt(nIndex:Integer):Integer;overload;
function FieldAsInt(sFieldName:String):Integer;overload;
function FieldAsFloat(nIndex:Integer):real;overload;
function FieldAsFloat(sFieldName:String):real;overload;
function FieldAsDatetime(nIndex:Integer):TDatetime;overload;
```

对查询结果记录操作的函数：

```
function ReordCount:Integer;
procedure NextRecord;
function EndOfQuery:boolean;
function BeginTrans:boolean;
function CommitTrans:boolean;
function RollbackTrans:boolean;
```

3. 4 本章小结

本章主要从总体功能模式出发，分析了各个功能模块的内容，以及各个模块实现方式等进行了详细的介绍。接口分别从数据库设计，现场总线设计、程序具体开发代码介绍了系统的设计过程。

第四章 系统功能实现

4.1 消防预警实现

4.1.1 底层温度信息采集

本文应用 LonWorks 现场总线技术和单总线数字温度传感器 DS18B20, 实现底层温度数据采集。

(1) 网络结构实现

楼宇自动化温度测控系统的网络结构中。LonWorks 控制系统网络由智能节点组成, 节点包括神经元芯片、传感器、控制设备、收发器和电源等。节点之间通信支持双绞线、电力线、光纤和红外线等多种介质, 按照规范的 Lon Talk 协议进行通信, 其通信速率范围在 $300 \text{ bp s} \sim 1.5 \text{ Mbp s}$ 之间。Neuron 芯片是 LonWorks 的核心, 它既进行通信管理, 也同时具有输入、输出和控制的能力。系统选用两级计算机监控系统, 即由中央监控 PC 机、Lon 网络适配卡以及多个智能节点组成。中央 PC 机控制节点的接口采用 Echelon 的 PCLTA - 10PCLonTalk 适配卡, 该卡是高性能的 16 位 ISA 总线 LonWorks 接口卡, 系统中, 通信介质为双绞线。网络采用基于 LonWorks 总线的网络模型, 节点数量可根据监控的需要进行开放增减。网络拓扑结构采用总线方式, 通信速率设为 78.125 kbp s 时, LonWorks 总线任意两节点之间的通信距离可以达到 2700m , 完全可以满足楼宇自动化系统的通信要求。中央监控 PC 机通过 Lon 网络适配卡与 LonWorks 总线相连, 用于整个系统的集中监控、管理、分析及网络通信检测等。

(2) 智能温度采集实现

系统采用基于主机的 LonWorks 智能节点, 选用美国 ATMEL 公司增强型 Flash 单片机 AT89S52 作为主处理器以完成主要的测控任务, 其内嵌 8 k FlashROM, 软硬件上兼容 AT89C52, 但其最大的特点是集成了 ISP 接口, 可直接在目标板上进行系统编程, 为用户带来了极大的方便; 单总线上挂接的 DS18B20 采用外接 VCC 方式而未用寄生供电, 以便除了正常测量各点室温外, 还可在火灾初期等异常情况下能准确工作; 利用 8155 扩展 I/O, 以对显示、键盘、超温报警等电路进行接口, 此外, 还通过

温控输出单元对空调机组进行新回风、送排风、喷淋管等阀门进行控制,达到控温的目的;Neuron 芯片采用美国 CY2PRESS 公司的 CY7C53120, 以其为核心, 再通过收发器 FTT- 10A 完成 LonTalk 协议的数据传输,并通过事件调度完成用户定义的各种计算、I/O 事件处理及网络报文处理等功能;收发器通过与 Lon 网接口负责将节点连入网络。单片机 AT89S52 与 Neuron 芯片 CY7C53120 采用并行通信。P1 口与 3120 的 I00~ I07 相连作为 8 位的数据总线。P3.2 与 3120 的 I08 相连,作为单片机请求发送数据的信号线和接受 3120 温度转换命令的应答线。P3.3 与 IO9 相连,作为 3120 接收数据的应答信号。P3.4 与 IO10 相连,作为 3120 发送温度转换命令的信号线。这就保证了 AT89S52 与 3120 通信的严格同步。

4.1.2 消防预警软件实现

4.1.2.1 主程序流程

主程序流程图如图 4.1 所示:

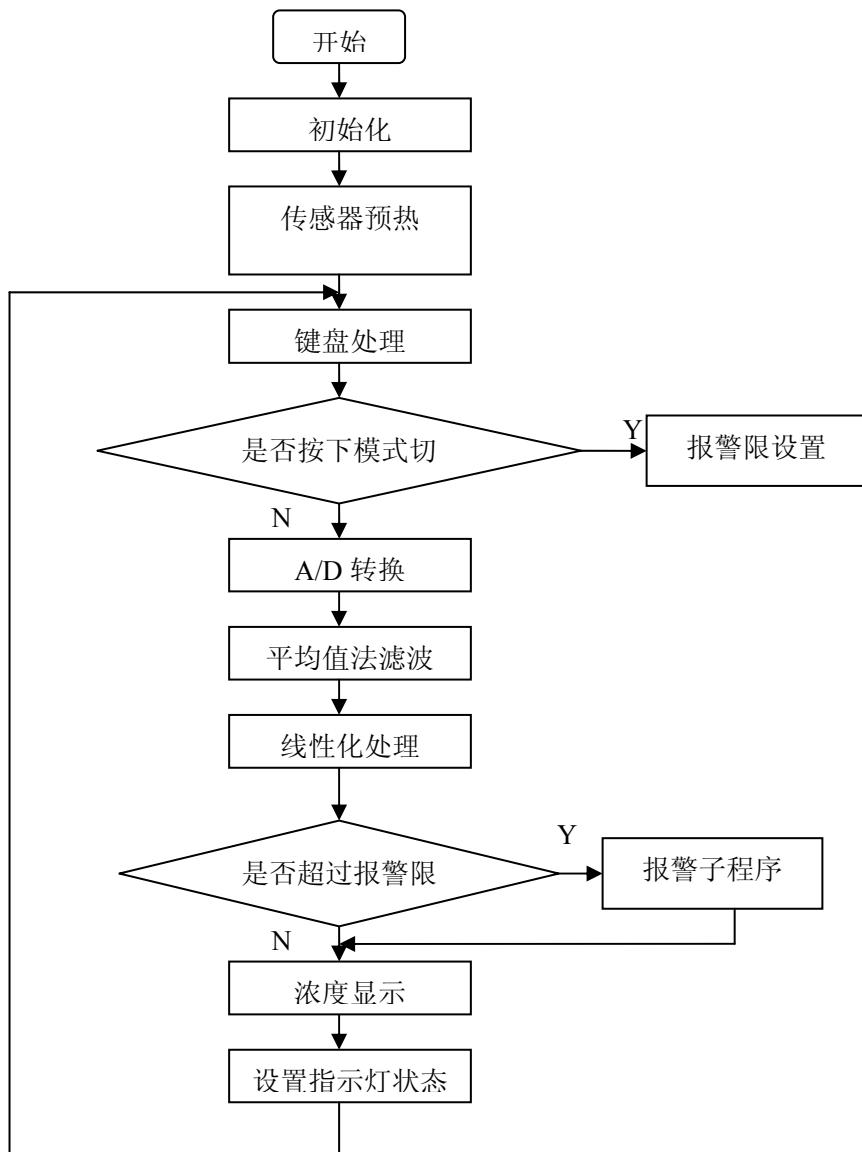


图 4-1 主程序流程图

系统启动后首先对楼宇中的温度传感器和烟雾传感器是否运行正常等进行判断，一般需要 3 分钟的检测试剂，若出现运行异常则前台告警，并提示运行异常设备位置。若设备运行正常则开始进入程序初始化。初始化结束后系统进入监控状态，AT89S52 单片机开始对楼宇中的烟雾浓度和温度信号进行 A/D 转换，然后将检测值与报警限设定值相比较，判断是否报警。

4.1.2.2 主程序初始化流程

系统硬件运行状态检测完毕后，开始进行系统初始化流程，其初始化流程如图 4-2 所示。其中功能实现部分包括 I/O 输入输出状态的设定、寄存器初始化、中

断使能等，通过设定定时器工作方式，然后开系统中断，以便响应中断定时，及时对气体浓度和温度进行采样。然后关闭蜂鸣器，开启绿灯，设置报警限初值。

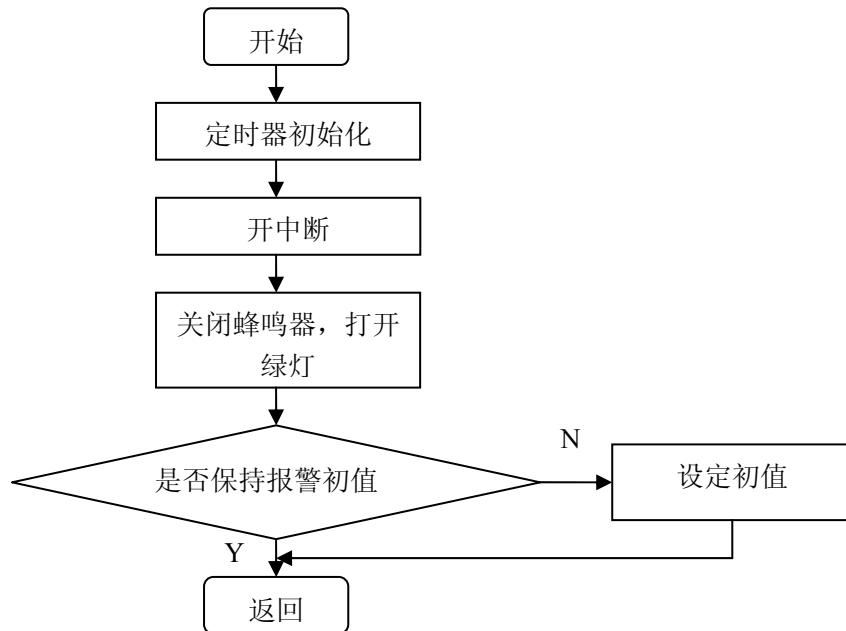


图 4-2 主程序初始化流程图

4.1.2.3 报警子程序

当烟雾浓度或温度值超过报警限设定值时，蜂鸣器发声，对应通道的红灯闪亮，以提示操作人员采取安全对策或自动控制相关安全装置，从而保障生产安全，避免火灾和爆炸事故的发生。报警子程序流程图如图 4-3 所示。

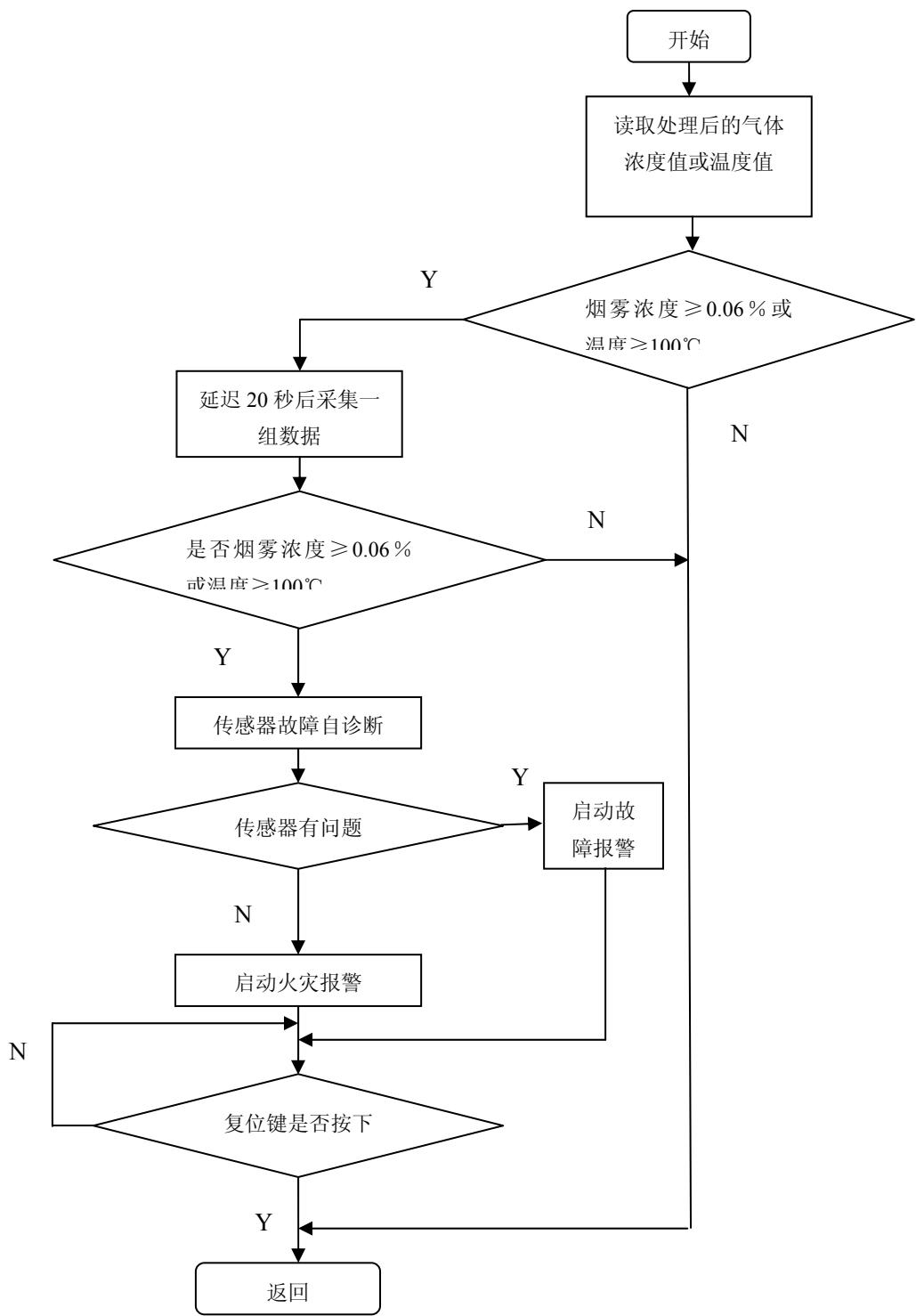


图 4-3 报警子程序流程图

4.2 系统功能页面展示

4.2.1 系统运行日志界面

系统日志界面主要是保存楼宇相关设备运行状态日志信息、历史记录、设备运行状态告警记录、以及设备异常关闭记录等。总的来说系统运行日志主要保存系统启动、关闭、告警、远程系统连接记录，它可以根据设备类型、时间、事件类型以及告警类型进行查询，同时在“设置事件”中还可以设置要日志系统中要记录的日志类型等。其具体界面如图 4-4 所示。

| 序号 | 事件描述 | 时间 |
|----|----------------------|---------------------|
| 1 | 启动监控 | 2007/10/19 14:13:20 |
| 2 | 初始化DSP失败 | 2007/10/19 14:13:21 |
| 3 | 发电机未授权监测，被忽略 | 2007/10/19 14:13:23 |
| 4 | WINDOW服务器未授权监测 | 2007/10/19 14:13:23 |
| 5 | UNIX服务器未授权监测，被忽略 | 2007/10/19 14:13:23 |
| 6 | Cisco3600路由器未授权监测 | 2007/10/19 14:13:23 |
| 7 | Catalyst5500交换机未授权监测 | 2007/10/19 14:13:23 |
| 8 | BOPHONE.DLL加载未成功 | 2007/10/19 14:13:24 |
| 9 | 短信COM2端口未能正常启动 | 2007/10/19 14:13:24 |
| 10 | COM1未能正常启动 | 2007/10/19 14:13:25 |
| 11 | 管理员登录系统 | 2007/10/19 14:13:28 |
| 12 | 退出监控 | 2007/10/19 14:28:07 |
| 13 | 启动监控 | 2007/10/19 14:28:25 |
| 14 | 初始化DSP失败 | 2007/10/19 14:28:25 |
| 15 | BOPHONE.DLL加载未成功 | 2007/10/19 14:28:29 |
| 16 | 短信COM2端口未能正常启动 | 2007/10/19 14:28:30 |
| 17 | COM1未能正常启动 | 2007/10/19 14:28:33 |
| 18 | 管理员登录系统 | 2007/10/19 14:28:33 |
| 19 | 退出监控 | 2007/10/19 14:29:28 |

图 4-4 历史事件查询

同时在事件查询页面中还有到处数据的功能，可以以 word 方式形成日志报告，也可以以 excel 方式导出事件详细清单，这些导出的文件可以用于系统日志检查报告或者导入到其它系统中。系统内部也附带着报表，以及报表扩展功能，通过报表扩展功能模块，用户可以根据自己的需求重新修改和定制新的报表。

4.2.2 自动化系统配置界面

在系统综合配置界面中，系统主要提供的是告警、设备告警级别、要设置告警的设备等监测点设置，通过界面中的“系统配置”模块来完成相关设置，其页面如 4-5 图所示：

| 选择设备 | | 所有设备 | 选择对象 | 设备配置 | 语音 | 更新 |
|------|--------------|---------|-------|--|-----|----|
| 序号 | 设备名称 | 通讯 | 设备配置 | 通讯故障处理提示 | | |
| 1 | 闭路监控 | 工作中断 | 模拟量配置 | 请检查 | EQ: | |
| 2 | 楼宇自控-空调机组 | 通讯故障 | 数字量配置 | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 3 | 楼宇自控-新风机组 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 4 | 楼宇自控-风机管盘 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 5 | 楼宇自控-照明 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 6 | 楼宇自控-电梯 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 7 | 楼宇自控-给排水 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 8 | 楼宇自控-透排水 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 9 | 楼宇自控-冷热源 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 10 | 防盗报警 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 11 | 消防报警 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 12 | 电子巡更 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 13 | STULZ空调 | 通讯故障 | | 1.电源掉电。2.通讯板故障。3.通讯线路故障。4.监控... EQ: | EQ: | |
| 14 | BA系统-精密空调 | 通讯错 | | 请查看通讯线路 | EQ: | |
| 15 | 楼宇自控-变配电 | 通讯故障 | | 1.电量仪掉电。2.通讯线故障。3.电源连接线故障。4.... EQ: | EQ: | |
| 16 | 发电机 | 通讯故障 | | 1.掉电。2.通讯线故障。3.电源连接线故障。4.监控机... EQ: | EQ: | |
| 17 | 门禁管理 | 控制器通讯故障 | | 1.掉电。2.通讯线故障。3.电源连接线故障。4.监控机... EQ: | EQ: | |
| 18 | WINDOW服务器 | 通讯中断 | | 请检查 | EQ: | |
| 19 | LINUX服务器 | 通讯中断 | | 请检查 | EQ: | |
| 20 | Cisco3600路由器 | 通讯中断 | | 请检查 | EQ: | |

图 4-5 系统综合配置

4. 2. 3 报警管理排表界面

在报警管理排表界面中主要负责人员设备管理。可以通过页面进行设备分类，以及管理员设备管理归类，以及管理员设备管理级别设置。超级管理员设置底下每个管理员所负责设备后，各自管理员只可以看到各自的设备信息，并进行监控。同时在该界面可以设置各自负责设备的告警时段、告警阀值、告警信息短信提醒或者电话通知等。页面提供了多个层级管理员人共同设备监控与管理等。同时界面中可以安排值班人员，即进行管理员值班情况记录与排值班表等，包括假期特殊值班排表等。其界面效果如图 4-6 所示：

| 报警人员值班表 | | | | | |
|---------|--------|---------|----------------------|-----------------|------|
| 设备分组: | | 设置人员信息: | | 分组报警负责人: | |
| ID | 名称 | 姓名 | 电话 | 短信 | Mail |
| 1 | 所有设备 | 李志强 | 80256835138003489... | aa1233@163.com | |
| 2 | BA组 | 刘洪 | 83221234138478902... | luhong@scnu.com | |
| 3 | 空调组 | 王冰 | 83524509136889345... | bw@126.com | |
| 4 | 配电组 | | | | |
| 5 | 消防组 | | | | |
| 6 | 楼宇组 | | | | |
| 7 | 电梯组 | | | | |
| 8 | 水泵组 | | | | |
| 9 | 冷水机组 | | | | |
| 10 | 热泵机组 | | | | |
| 11 | 变频器组 | | | | |
| 12 | 水泵控制组 | | | | |
| 13 | 冷水机控制组 | | | | |
| 14 | 热泵机控制组 | | | | |
| 15 | 变频器控制组 | | | | |
| 16 | 水泵控制组 | | | | |
| 17 | 冷水机控制组 | | | | |
| 18 | 热泵机控制组 | | | | |
| 19 | 变频器控制组 | | | | |
| 20 | 水泵控制组 | | | | |
| 21 | 冷水机控制组 | | | | |
| 22 | 热泵机控制组 | | | | |
| 23 | 变频器控制组 | | | | |
| 24 | 水泵控制组 | | | | |
| 25 | 冷水机控制组 | | | | |
| 26 | 热泵机控制组 | | | | |
| 27 | 变频器控制组 | | | | |
| 28 | 水泵控制组 | | | | |
| 29 | 冷水机控制组 | | | | |
| 30 | 热泵机控制组 | | | | |
| 31 | 变频器控制组 | | | | |
| 32 | 水泵控制组 | | | | |
| 33 | 冷水机控制组 | | | | |
| 34 | 热泵机控制组 | | | | |
| 35 | 变频器控制组 | | | | |
| 36 | 水泵控制组 | | | | |
| 37 | 冷水机控制组 | | | | |
| 38 | 热泵机控制组 | | | | |
| 39 | 变频器控制组 | | | | |
| 40 | 水泵控制组 | | | | |
| 41 | 冷水机控制组 | | | | |
| 42 | 热泵机控制组 | | | | |
| 43 | 变频器控制组 | | | | |
| 44 | 水泵控制组 | | | | |
| 45 | 冷水机控制组 | | | | |
| 46 | 热泵机控制组 | | | | |
| 47 | 变频器控制组 | | | | |
| 48 | 水泵控制组 | | | | |
| 49 | 冷水机控制组 | | | | |
| 50 | 热泵机控制组 | | | | |
| 51 | 变频器控制组 | | | | |
| 52 | 水泵控制组 | | | | |
| 53 | 冷水机控制组 | | | | |
| 54 | 热泵机控制组 | | | | |
| 55 | 变频器控制组 | | | | |
| 56 | 水泵控制组 | | | | |
| 57 | 冷水机控制组 | | | | |
| 58 | 热泵机控制组 | | | | |
| 59 | 变频器控制组 | | | | |
| 60 | 水泵控制组 | | | | |
| 61 | 冷水机控制组 | | | | |
| 62 | 热泵机控制组 | | | | |
| 63 | 变频器控制组 | | | | |
| 64 | 水泵控制组 | | | | |
| 65 | 冷水机控制组 | | | | |
| 66 | 热泵机控制组 | | | | |
| 67 | 变频器控制组 | | | | |
| 68 | 水泵控制组 | | | | |
| 69 | 冷水机控制组 | | | | |
| 70 | 热泵机控制组 | | | | |
| 71 | 变频器控制组 | | | | |
| 72 | 水泵控制组 | | | | |
| 73 | 冷水机控制组 | | | | |
| 74 | 热泵机控制组 | | | | |
| 75 | 变频器控制组 | | | | |
| 76 | 水泵控制组 | | | | |
| 77 | 冷水机控制组 | | | | |
| 78 | 热泵机控制组 | | | | |
| 79 | 变频器控制组 | | | | |
| 80 | 水泵控制组 | | | | |
| 81 | 冷水机控制组 | | | | |
| 82 | 热泵机控制组 | | | | |
| 83 | 变频器控制组 | | | | |
| 84 | 水泵控制组 | | | | |
| 85 | 冷水机控制组 | | | | |
| 86 | 热泵机控制组 | | | | |
| 87 | 变频器控制组 | | | | |
| 88 | 水泵控制组 | | | | |
| 89 | 冷水机控制组 | | | | |
| 90 | 热泵机控制组 | | | | |
| 91 | 变频器控制组 | | | | |
| 92 | 水泵控制组 | | | | |
| 93 | 冷水机控制组 | | | | |
| 94 | 热泵机控制组 | | | | |
| 95 | 变频器控制组 | | | | |
| 96 | 水泵控制组 | | | | |
| 97 | 冷水机控制组 | | | | |
| 98 | 热泵机控制组 | | | | |
| 99 | 变频器控制组 | | | | |
| 100 | 水泵控制组 | | | | |
| 101 | 冷水机控制组 | | | | |
| 102 | 热泵机控制组 | | | | |
| 103 | 变频器控制组 | | | | |
| 104 | 水泵控制组 | | | | |
| 105 | 冷水机控制组 | | | | |
| 106 | 热泵机控制组 | | | | |
| 107 | 变频器控制组 | | | | |
| 108 | 水泵控制组 | | | | |
| 109 | 冷水机控制组 | | | | |
| 110 | 热泵机控制组 | | | | |
| 111 | 变频器控制组 | | | | |
| 112 | 水泵控制组 | | | | |
| 113 | 冷水机控制组 | | | | |
| 114 | 热泵机控制组 | | | | |
| 115 | 变频器控制组 | | | | |
| 116 | 水泵控制组 | | | | |
| 117 | 冷水机控制组 | | | | |
| 118 | 热泵机控制组 | | | | |
| 119 | 变频器控制组 | | | | |
| 120 | 水泵控制组 | | | | |
| 121 | 冷水机控制组 | | | | |
| 122 | 热泵机控制组 | | | | |
| 123 | 变频器控制组 | | | | |
| 124 | 水泵控制组 | | | | |
| 125 | 冷水机控制组 | | | | |
| 126 | 热泵机控制组 | | | | |
| 127 | 变频器控制组 | | | | |
| 128 | 水泵控制组 | | | | |
| 129 | 冷水机控制组 | | | | |
| 130 | 热泵机控制组 | | | | |
| 131 | 变频器控制组 | | | | |
| 132 | 水泵控制组 | | | | |
| 133 | 冷水机控制组 | | | | |
| 134 | 热泵机控制组 | | | | |
| 135 | 变频器控制组 | | | | |
| 136 | 水泵控制组 | | | | |
| 137 | 冷水机控制组 | | | | |
| 138 | 热泵机控制组 | | | | |
| 139 | 变频器控制组 | | | | |
| 140 | 水泵控制组 | | | | |
| 141 | 冷水机控制组 | | | | |
| 142 | 热泵机控制组 | | | | |
| 143 | 变频器控制组 | | | | |
| 144 | 水泵控制组 | | | | |
| 145 | 冷水机控制组 | | | | |
| 146 | 热泵机控制组 | | | | |
| 147 | 变频器控制组 | | | | |
| 148 | 水泵控制组 | | | | |
| 149 | 冷水机控制组 | | | | |
| 150 | 热泵机控制组 | | | | |
| 151 | 变频器控制组 | | | | |
| 152 | 水泵控制组 | | | | |
| 153 | 冷水机控制组 | | | | |
| 154 | 热泵机控制组 | | | | |
| 155 | 变频器控制组 | | | | |
| 156 | 水泵控制组 | | | | |
| 157 | 冷水机控制组 | | | | |
| 158 | 热泵机控制组 | | | | |
| 159 | 变频器控制组 | | | | |
| 160 | 水泵控制组 | | | | |
| 161 | 冷水机控制组 | | | | |
| 162 | 热泵机控制组 | | | | |
| 163 | 变频器控制组 | | | | |
| 164 | 水泵控制组 | | | | |
| 165 | 冷水机控制组 | | | | |
| 166 | 热泵机控制组 | | | | |
| 167 | 变频器控制组 | | | | |
| 168 | 水泵控制组 | | | | |
| 169 | 冷水机控制组 | | | | |
| 170 | 热泵机控制组 | | | | |
| 171 | 变频器控制组 | | | | |
| 172 | 水泵控制组 | | | | |
| 173 | 冷水机控制组 | | | | |
| 174 | 热泵机控制组 | | | | |
| 175 | 变频器控制组 | | | | |
| 176 | 水泵控制组 | | | | |
| 177 | 冷水机控制组 | | | | |
| 178 | 热泵机控制组 | | | | |
| 179 | 变频器控制组 | | | | |
| 180 | 水泵控制组 | | | | |
| 181 | 冷水机控制组 | | | | |
| 182 | 热泵机控制组 | | | | |
| 183 | 变频器控制组 | | | | |
| 184 | 水泵控制组 | | | | |
| 185 | 冷水机控制组 | | | | |
| 186 | 热泵机控制组 | | | | |
| 187 | 变频器控制组 | | | | |
| 188 | 水泵控制组 | | | | |
| 189 | 冷水机控制组 | | | | |
| 190 | 热泵机控制组 | | | | |
| 191 | 变频器控制组 | | | | |
| 192 | 水泵控制组 | | | | |
| 193 | 冷水机控制组 | | | | |
| 194 | 热泵机控制组 | | | | |
| 195 | 变频器控制组 | | | | |
| 196 | 水泵控制组 | | | | |
| 197 | 冷水机控制组 | | | | |
| 198 | 热泵机控制组 | | | | |
| 199 | 变频器控制组 | | | | |
| 200 | 水泵控制组 | | | | |
| 201 | 冷水机控制组 | | | | |
| 202 | 热泵机控制组 | | | | |
| 203 | 变频器控制组 | | | | |
| 204 | 水泵控制组 | | | | |
| 205 | 冷水机控制组 | | | | |
| 206 | 热泵机控制组 | | | | |
| 207 | 变频器控制组 | | | | |
| 208 | 水泵控制组 | | | | |
| 209 | 冷水机控制组 | | | | |
| 210 | 热泵机控制组 | | | | |
| 211 | 变频器控制组 | | | | |
| 212 | 水泵控制组 | | | | |
| 213 | 冷水机控制组 | | | | |
| 214 | 热泵机控制组 | | | | |
| 215 | 变频器控制组 | | | | |
| 216 | 水泵控制组 | | | | |
| 217 | 冷水机控制组 | | | | |
| 218 | 热泵机控制组 | | | | |
| 219 | 变频器控制组 | | | | |
| 220 | 水泵控制组 | | | | |
| 221 | 冷水机控制组 | | | | |
| 222 | 热泵机控制组 | | | | |
| 223 | 变频器控制组 | | | | |
| 224 | 水泵控制组 | | | | |
| 225 | 冷水机控制组 | | | | |
| 226 | 热泵机控制组 | | | | |
| 227 | 变频器控制组 | | | | |
| 228 | 水泵控制组 | | | | |
| 229 | 冷水机控制组 | | | | |
| 230 | 热泵机控制组 | | | | |
| 231 | 变频器控制组 | | | | |
| 232 | 水泵控制组 | | | | |
| 233 | 冷水机控制组 | | | | |
| 234 | 热泵机控制组 | | | | |
| 235 | 变频器控制组 | | | | |
| 236 | 水泵控制组 | | | | |
| 237 | 冷水机控制组 | | | | |
| 238 | 热泵机控制组 | | | | |
| 239 | 变频器控制组 | | | | |
| 240 | 水泵控制组 | | | | |
| 241 | 冷水机控制组 | | | | |
| 242 | 热泵机控制组 | | | | |
| 243 | 变频器控制组 | | | | |
| 244 | 水泵控制组 | | | | |
| 245 | 冷水机控制组 | | | | |
| 246 | 热泵机控制组 | | | | |
| 247 | 变频器控制组 | | | | |
| 248 | 水泵控制组 | | | | |
| 249 | 冷水机控制组 | | | | |
| 250 | 热泵机控制组 | | | | |
| 251 | 变频器控制组 | | | | |
| 252 | 水泵控制组 | | | | |
| 253 | 冷水机控制组 | | | | |
| 254 | 热泵机控制组 | | | | |
| 255 | 变频器控制组 | | | | |
| 256 | 水泵控制组 | | | | |
| 257 | 冷水机控制组 | | | | |
| 258 | 热泵机控制组 | | | | |
| 259 | 变频器控制组 | | | | |
| 260 | 水泵控制组 | | | | |
| 261 | 冷水机控制组 | | | | |
| 262 | 热泵机控制组 | | | | |
| 263 | 变频器控制组 | | | | |
| 264 | 水泵控制组 | | | | |
| 265 | 冷水机控制组 | | | | |
| 266 | 热泵机控制组 | | | | |
| 267 | 变频器控制组 | | | | |
| 268 | 水泵控制组 | | | | |
| 269 | 冷水机控制组 | | | | |
| 270 | 热泵机控制组 | | | | |
| 271 | 变频器控制组 | | | | |
| 272 | 水泵控制组 | | | | |
| 273 | 冷水机控制组 | | | | |
| 274 | 热泵机控制组 | | | | |
| 275 | 变频器控制组 | | | | |
| 276 | 水泵控制组 | | | | |
| 277 | 冷水机控制组 | | | | |
| 278 | 热泵机控制组 | | | | |
| 279 | 变频器控制组 | | | | |
| 280 | 水泵控制组 | | | | |
| 281 | 冷水机控制组 | | | | |
| 282 | 热泵机控制组 | | | | |
| 283 | 变频器控制组 | | | | |
| 284 | 水泵控制组 | | | | |
| 285 | 冷水机控制组 | | | | |
| 286 | 热泵机控制组 | | | | |
| 287 | 变频器控制组 | | | | |
| 288 | 水泵控制组 | | | | |
| 289 | 冷水机控制组 | | | | |

```

Private Void CallAlarmService0

{
    AsyncCallbackAsyncCallbaek=new AsyncCallback( MyCaUBack);

    //创建建调用结束回撤委托对象

    IAsyncResultAsyncResult;

    AsyncResult=MyService. BeginGetAlarm(AsyncResult, null);

    //开始调用

}

Private void MyCallBack(System. IAsyncResultAsyncResult) / / 回撤函数

(
    boolRetumValue;

    RetumValue=MyService. EndGetAlarm(AsyncResult); / / 结束调用

    if(RemmvValue==ture)

    (
        DealWithAlarm(); / / 报警处理

    )

}

```

4. 2. 5 综合保安管理界面



图 4-8 视频监控模块

视频监控模块支持个性化窗口定义、历史数据查询、进行云控制等，同时可以进行远程管理与控制。

其部分代码：

```
#define FLEFTBAR      1    //左边框
#define FLEFTTITLE     2    //左标题
#define FRIGHTTITLE    4    //右标题
#define FMIDTITLE      8    //中间标题
#define FRIGHTBAR      16   //右边框
#define FBOTTOMBAR     32   //底边框
#define FMINBUTTON     64   //最小化按钮
#define FMAXBUTTON     128  //最大化按钮
#define FCLOSEBUTTON   256  //关闭按钮
#define FALL           511  //所有标识
#define MAXNUM         100

//按钮状态
Onlintdialog
//加载背景位图
m_BKGround.LoadBitmap(IDB_BKGROUN);
m_BorderHeight = GetSystemMetrics(SM_CYBORDER);
m_BorderWidth = GetSystemMetrics(SM_CXBORDER);
m_CaptionHeight = GetSystemMetrics(SM_CYCAPTION);
SetWindowText(m_Caption);
m_CaptionFont.CreateFont(14,10,0,3,600,0,0,0,ANSI_CHARSET,OUT_DEFAULT_PRECIS,
CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,FF_ROMAN,"宋体");
CBitmap bitmap;
bitmap.LoadBitmap(IDB_MINBT);
BITMAPINFO bInfo;
bitmap.GetObject(sizeof(bInfo),&bInfo);
m_ButtonWidth = bInfo.bmiHeader.biWidth;
m_ButtonHeight = bInfo.bmiHeader.biHeight;
bitmap.DeleteObject();
-----
```

4. 2. 6 车库收费管理

(2) 车型管理：用户修改车型

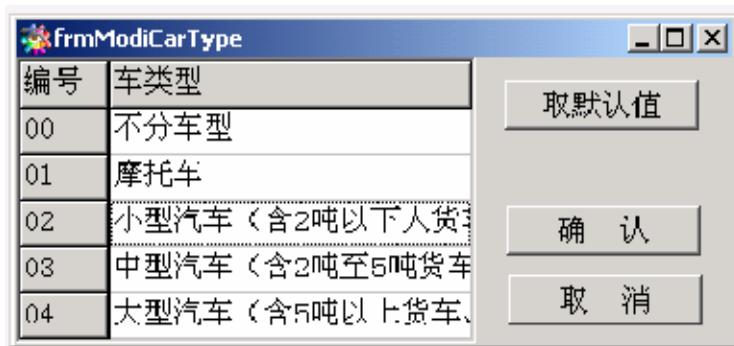


图 4-9 车型管理

在停车场收费管理中提供了收费标准设置、收费方式和临时收费标准三种类型，在标准收费标准页面中可以设置该停车场目前的停车费标准；收费方式可以设置用户属于包月还是零食，还是一次性交费方式管理；在临时收费标准设置模块中，可以进行分时段不同收费标准设置，还可以根据车型进行另外收费等。其设置页面如图 4-10 所示



图 4-10 车型费用管理

在页面中可以提供了用户管理，通过用户管理可以看出该用户所拥有车型的车牌号、车型、交费方式、违规记录等。同时页面提供收费对账功能，用户和用户进行结算稽核等。

4.3 消防预警温度信息 Web 查询

本论文选择的楼宇为国际航站作为选择案例，对其楼宇消防预警系统、安保系统等智能化楼宇设计。在国际航站楼宇中，区域温度查询程序分为 3 个页面，温度实时查询，修正温度查询更改和日温度曲线图。温度实时查询与以上两个查

询类似，还不需要对数据作特殊显示处理，故代码相对简单。而修正温度查询更改完全由 gridview 控件自带功能实现，直接与数据表绑定，代码更为简单。日温 度曲线使用了 utralchart 控件，并自定义了相对的 chart 生成函数。具体界面和代码 分别如下：

```
protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    GridView1.PageIndex = e.NewPageIndex;
    databindtoGV(); //重新绑定 GridView 数据的函数
    fontcolor();
}

protected void Page_Load(object sender, EventArgs e) //页面数据加载函数
{
    if (!IsPostBack)
    {
        HtmlMeta tag = new HtmlMeta();
        tag.HttpEquiv = "refresh";
        tag.Content = "60";
        Header.Controls.Add(tag);
        databindtoGV();
        fontcolor();
    }
}
```

当前位置：楼宇监控查询系统→暖通空调系统→区域温度实时查询

| 区域空调实时查询 | | | |
|----------|----------|---------|---------------------|
| 区域代码 | 区域名称 | 大厅温度(℃) | 记录时间 |
| 0001 | 国际到达一楼 | 18.3℃ | 2007-11-23 14:19:03 |
| 0002 | 国内到达一楼 | 16.9℃ | 2007-11-23 14:19:03 |
| 0003 | 国内到达二楼 | 20.4℃ | 2007-11-23 14:19:03 |
| 0004 | 国际到达二楼 | 18.8℃ | 2007-11-23 14:19:03 |
| 0005 | 国内出发大厅三楼 | 21.3℃ | 2007-11-23 14:19:03 |
| 0006 | 国际出发大厅三楼 | 18.0℃ | 2007-11-23 14:19:03 |
| 0007 | 国内隔离厅三楼 | 23.2℃ | 2007-11-23 14:19:03 |
| 0008 | 国际隔离厅三楼 | 23.9℃ | 2007-11-23 14:19:03 |
| 0009 | 航站楼五楼 | 20.2℃ | 2007-11-23 14:19:03 |

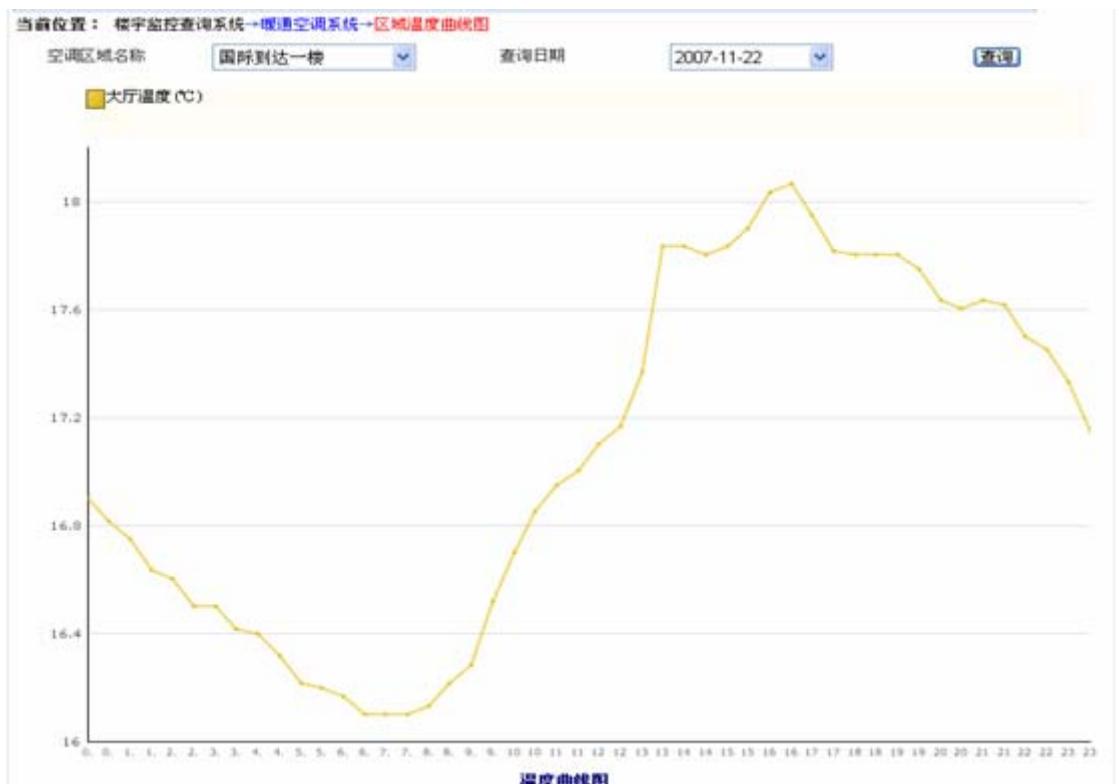


图 4-11 温度信息 Web 实时查询

其温度曲线设计想代码为：

```

protected void Page_Load(object sender, EventArgs e)
{
    .....
    {
        DropDownList1.SelectedValue = Request.QueryString["id"];
        imgID = Request.QueryString["id"];
    }
    WebDateChooser1.Value = System.DateTime.Now.ToString("yyyy-MM-dd");//获取时间
    DataSetimgTableAdapters.aircondinfoTableAdapter imgApd = new
    DataSetimgTableAdapters.aircondinfoTableAdapter();
    DataSetimg.aircondinfoDataTable imgDt = new DataSetimg.aircondinfoDataTable();
    imgDt = imgApd.GetData(imgID,
    Convert.ToDateTime(WebDateChooser1.Value).ToString("yyyy-MM-dd"));
    UltraChart1.DataSource = imgDt;
    UltraChart1.DataBind(); //数据图提取函数
    initLineChart(UltraChart1, "温度曲线图", new int[] { }, "大厅温度(℃)",
    "<ITEM_LABEL>"); //数据页面展示函数
}
-----}

```

4. 4 本章小结

本章是系统主要从整个楼宇自动化控制系统实现界面进行说明。主要是进行系统完成后的部分功能截图展示。

第五章 系统运行测试

5.0 测试概述

(1) 测试内容

系统主要采用了基于 Web 的远程数据查询，因此我需要对 Web 服务系统进行相关测试，测试内容主要基于楼宇信息 web 查询服务的压力和性能测试。

- 压力测试：

在这部分测试中，我们会测试系统的在长时间运行情况下以及在多用户并发访问下的工作情况。我们要保证系统能达到一般流行论坛的级别，在长时间运行和高并发访问的一些极端情况下不会崩溃。

- 性能测试：

在这部分测试中，我们会测试系统的响应时间和资源消耗。我们要保证系统在网络正常时能提供一般网站访问的响应时间，这对于基于 web 访问的应用是至关重要的，因为如果响应时间过长，用户就会失去耐心，从而放弃使用。同时，我们要保证系统的资源消耗应该能处于服务器允许的正常阈值以下，这样服务器才能保证比较稳定的工作。

(2) 测试目标

网站系统支持 50 个以上并发客户端的访问。

(3) 技术目标

使用测试工具实现虚拟用户并发压力测试，要求系统满足用户并发量在 50 以上，并能正常工作。

5.1 测试环境

(1) 软硬件环境

| 设备名称 | 硬件配置 | 软件配置 | 备注 |
|------------|--|---|----|
| Web+数据库服务器 | | | |
| 负载生成器 | CPU: P9400 2.53GHz 内存: DDR3 512MB 硬盘: 30GB | 操作系统: Windows_XP_Sp3_OEM 负载生成工具: LoadRunner8.1 | |

(2) 网络结构

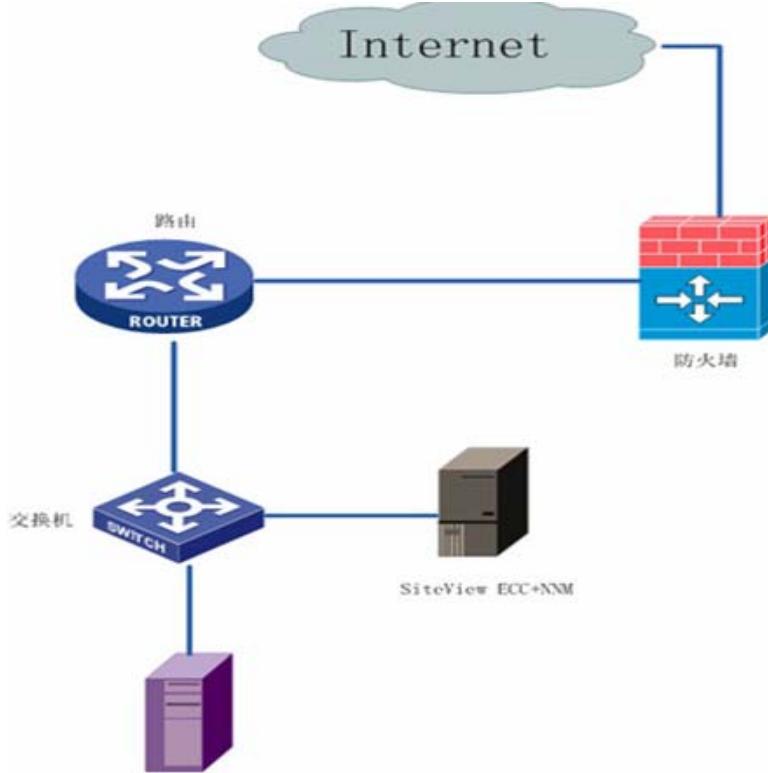


图 5-1 测试网络拓扑结构

5.2 测试场景与工具

(1) 测试场景

本次性能负载是采取使用手工场景来完成，设置为每 12 秒添加 2 个用户，一直增加负载至 50 个用户，持续 5 分钟后结束场景，设置用户为线程模式，宽带是 512kbps, 其他选项均为默认值。

(2) 测试要求与内容

根据需求，对登录操作进行并发的压力测试，对主要业务模块中的主要业务进行压力和负载测试。

(3) 测试通过标准

系统在并发用户 50 时，系统表现稳定。系统响应时间不超过 4s。

(4) 测试工具

LoadRunner 是一种预测系统行为和性能的负载测试工具。通过模拟上千用户实施并发负载及实时性能测试的方式来确认和查找问题，LoadRunner 能够对整个

企业架构进行测试。通过 LoadRunner，企业能最大限度的缩短测试时间、优化性能和加速应用系统的发布周期。

由于这里的测试目标是了解网站在负载下的情况进一步进行瓶颈分析，故使用 LoadRunner 的手工场景进行性能测试。

5.3 测试结果分析

5.3.1 分析摘要

(1) 分析摘要

LoadRunner 进行 50 个用户场景模拟测试结果收集后，显示的该结果的一个摘要信息，如图所示。概要中列出了场景执行情况、“Statistics Summary（统计信息摘要）”、“Transaction Summary（事务摘要）”以及“HTTP Responses Summary（HTTP 响应摘要）”等。以简要的信息列出本次测试结果。



图 5-2 分析摘要

该部分给出了本次测试场景的名称、结果存放路径及场景的持续时间，如上图所示。从该图我们知道，本次测试从 2012-03-11 00:59:24 开始，到 2012-03-11 01:09:47 结束，共历时 09 分 23 秒。

(2) 统计信息摘要

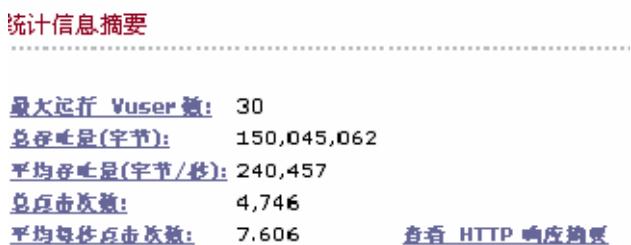


图 5-3 统计信息摘要

该部分给出了场景执行结束后并发数、总吞吐量、平均每秒吞吐量、总请求数、平均每秒请求数的统计值。从该图我们得知，本次测试运行的最大并发数为 50，总吞吐量为 150,045,062 字节，平均每秒的吞吐量为 240,457 字节，总的请求数为 4,746，平均每秒的请求为 52.024。

(3) 事务摘要

| 事务摘要 | | | | | | | | |
|-------------------------------|--------|--------|---------|--------|---------|-----|----|----|
| 业务: 通过总数: 214 失败总数: 0 停止总数: 0 | | | 平均响应时间 | | | | | |
| 事务名 | 最小值 | 平均值 | 最大值 | 标准偏差 | 99% | 通过 | 失败 | 停止 |
| Action_Transaction | 64.625 | 91.366 | 111.422 | 11.928 | 104.922 | 154 | 0 | 0 |
| vuser_end_Transaction | 6.672 | 7.409 | 8.594 | 0.382 | 7.906 | 30 | 0 | 0 |
| vuser_init_Transaction | 11.775 | 13.485 | 14.519 | 0.564 | 13.969 | 30 | 20 | 0 |

图 5-4 事物摘要

该部分给出了场景执行结束后相关 Action 的平均响应时间、通过率等情况，从该图我们得到每个 Action 的平均响应时间与业务成功率。

(4) HTTP 响应摘要

| HTTP 响应摘要 | | |
|-----------|-------|-------|
| HTTP 响应 | 总计 | 每秒 |
| HTTP_200 | 4,746 | 7.606 |

图 5-5HTTP 响应摘要

该部分显示在场景执行过程中，每次 HTTP 请求发出去的状态，是成功还是失败，都在这里体现，如图所示。从图中可以看到，在本次测试过程中 LoadRunner 共模拟发出了 4,767 次请求（与“统计信息摘要”中的“总点击次数”一致），其中“HTTP 200”的是 4,767 次，说明在本次过程中，经过发出的请求全部分都能正确响应了（“HTTP 200”表示请求被正确响应）。

5.3.2 运行 Vuser

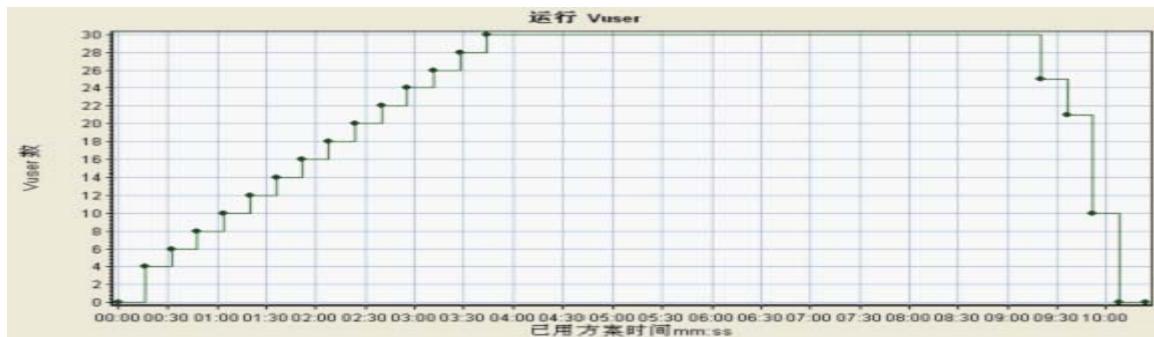


图 5-6 Vuser 运行状况

该图反应系统形成负载过程，随着时间的推移，虚拟用户逐渐增加，在 03:45 分时，达到了负载峰值 30 个虚拟用户，负载生成大约是每分钟增加 2 个用户，峰值负载持续 6 分钟。

5.3.3 平均事务响应时间

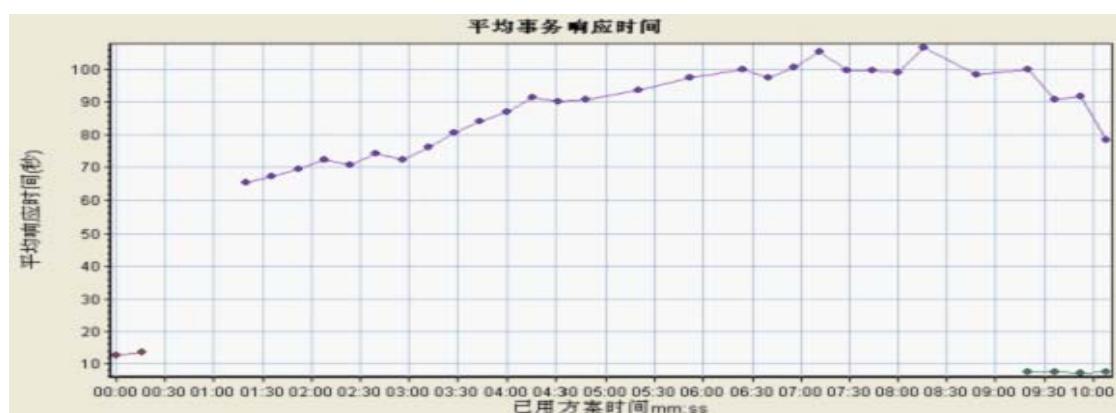


图 5-7 Vuser 平均事务响应时间

| 颜色 | 比例 | 度量 | 最 小 值 | 平 均 值 | 最 大 值 | SD |
|----|----|------------------------|--------|--------|---------|--------|
| ■ | 1 | Action_Transaction | 64.625 | 91.366 | 111.422 | 11.928 |
| ■ | 1 | vuser_end_Transaction | 6.672 | 7.409 | 8.594 | 0.382 |
| ■ | 1 | vuser_init_Transaction | 11.775 | 13.485 | 14.519 | 0.564 |

从图形我们可以看到 Action、vuser_end、vuser_init 三者的平均时间分别是

91.366 秒， 7.409 秒， 13.485 秒，与实际结果相差大。与网络、CPU、系统资源占用有关。

5.3.4 运行 Vuser-平均事务响应时间

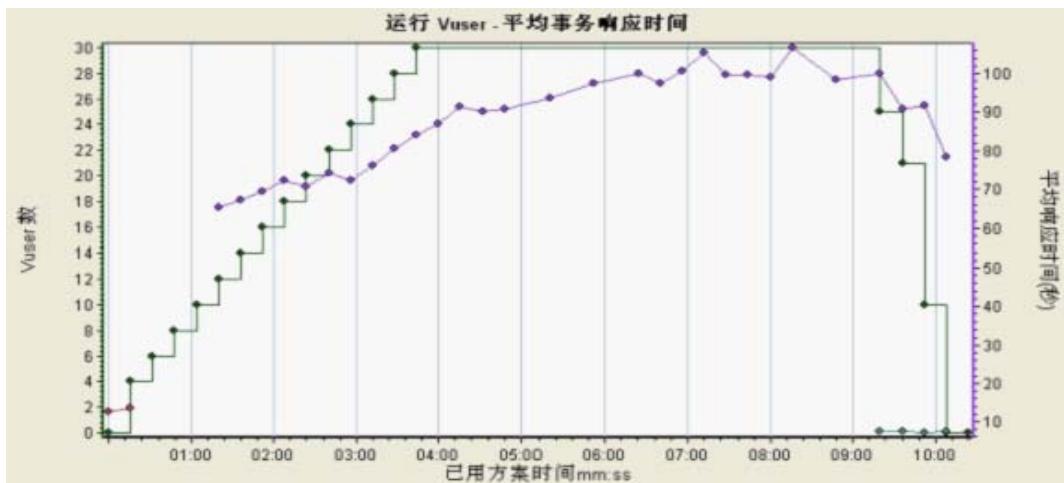


图 5-8 Vuser-平均事务响应时间

此图是 vuser 和平均事务响应时间合成图，从图中可以看出 vuser_init_Transaction（系统登陆）对系统毫无影响，vuser 数达到 17 个时，平均事务响应时间才有所增加，也就是说系统达到最优性能时能同时允许 17 个用户同时处理事务，用户达到 30 个，这时系统响应时间最大，而这个响应时间推迟 4 分钟后才出现，说明这时系统不能满足用户需求。

5.3.5 每秒点击数

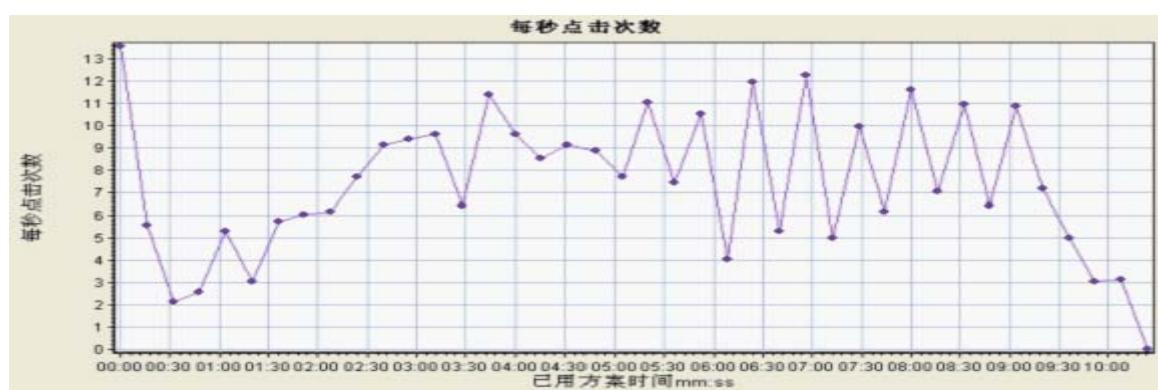


图 5-9 每秒点击数

每秒点击数提供当前负载中对系统所产生的点击量记录，每一次点击相当于对服务器发出一次请求，该数据越大越好，从图中可以看出，随着时间推移，点击数呈波浪形状，高低起伏大，说明网络服务器不能及时接收用户请求，最高达13次/s

5.3.6 每秒点击数-每秒HTTP请求数响应数

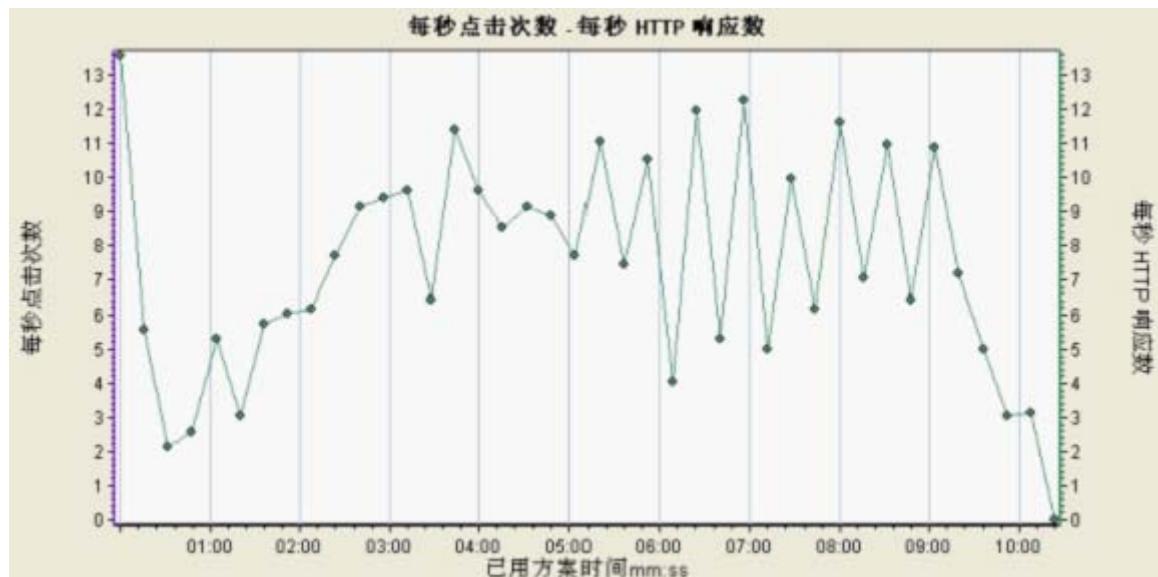


图 5-10 每秒 HTTP 响应数

| Color | Graph | Scale | Measurement | 图的最小值 | Graph's Average | 图的最大值 | 图的中值 | 图的 SD |
|-------|-------------|-------|-------------|-------|-----------------|--------|-------|-------|
| 紫色 | 每秒点击次数 | 1 | 点击次数 | 0.0 | 7.416 | 13.563 | 7.438 | 3.159 |
| 绿色 | 每秒 HTTP 响应数 | 1 | HTTP_200 | 0.0 | 7.416 | 13.563 | 7.438 | 3.159 |

点击数是指客户端发出请求数，而HTTP响应数是指服务器的返回响应数，2个数值一般都是相同，图中显示该数值都相同，说明服务器能答应超出负载的连接请求。

5.3.7 吞吐量-每秒点击数

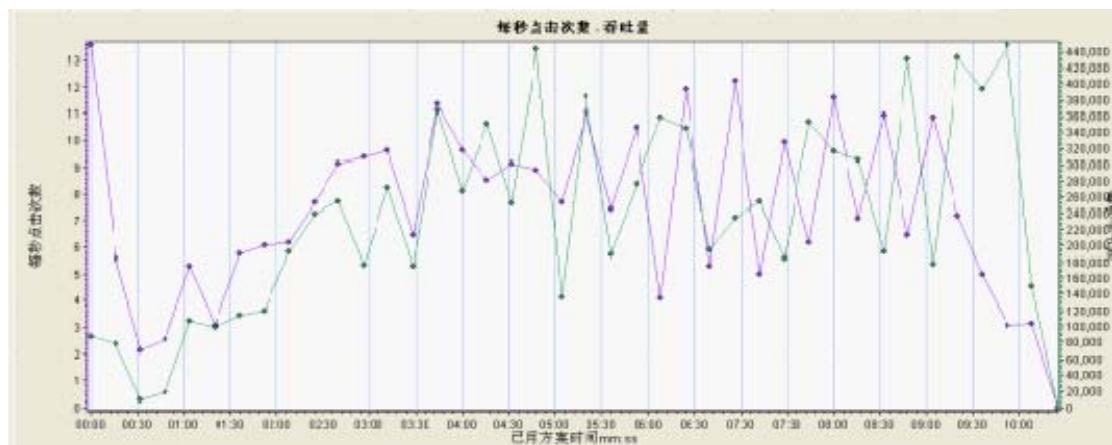


图 5-11 吞吐量-每秒点击数

从图中可以看出，两种图形的曲线都正常并且基本一致，说明服务器能及时的接受客户端的请求，并能够返回结果。

5.3.8 事务摘要

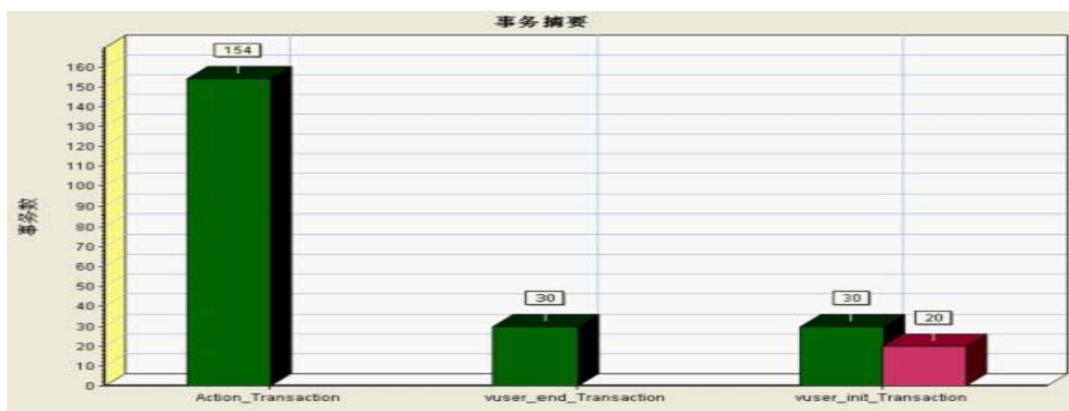


图 5-12 事物摘要

| 颜色 | 比例 | 度量 |
|----|----|------|
| ■ | 1 | Pass |
| ■ | 1 | Fail |

从图中可以看出，Action 登陆操作一共有 154 次成功，0 次失败，vuser_end 用户退出操作 30 个用户全部安全退出，vuser_int 一个 50 个用户登陆网站，30 个成功.，20 个失败。

5.3.9 事务性能摘要

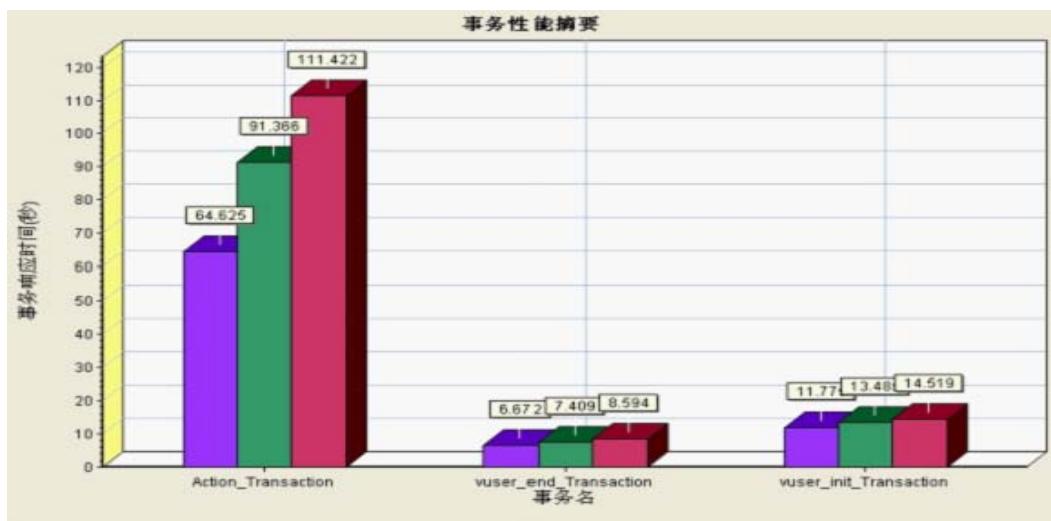


图 5-13 事务性能摘要

| 颜色 | 比例 | 度量 |
|----|----|-----|
| 紫色 | 1 | 最小值 |
| 绿色 | 1 | 平均值 |
| 红色 | 1 | 最大值 |

从图中可以看错，Action 事务最大值是 111.422s，平均值是 91.366s，最小值是 64.625s，相差大，说明此时系统不够稳定。而 vuser_end 最大值是 8.594s，平均值是 7.409s，最小值是 6.627s 和 vuser_init 最大值是 14.519s，平均值是 13.484s，最小值是 11.774s 都相差小，说明这时系统稳定。

5.3.10 Windows 资源

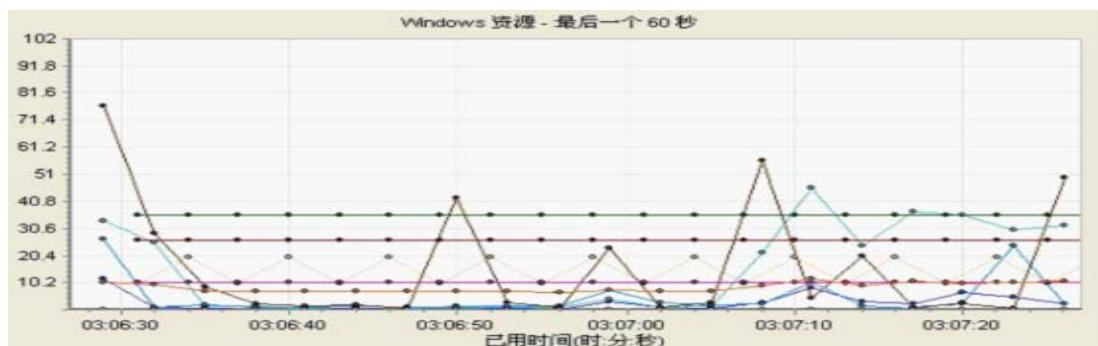


图 5-14 windows 资源使用情况

| 颜色 | 比例 | 度量 | 计算机 | 最大值 | 最小值 | 平均值 | 标准值 | 最后一个 |
|------|-------------------------------------|-----------|---------------|---------------|---------------|------------|---------------|------|
| 1 | % Processor Time (Processor _Total) | localhost | 11.458 | 0.000 | 2.629 | 3.148 | 2.083 | |
| 1 | File Data Operations/sec (System) | localhost | 47.003 | 0.333 | 16.493 | 16.943 | 32.001 | |
| 10 | Processor Queue Length (System) | localhost | 2.000 | 1.000 | 1.524 | 0.499 | 2.000 | |
| 0.1 | Page Faults/sec (Memory) | localhost | 267.606 | 5.006 | 45.273 | 71.719 | 22.001 | |
| 10 | % Disk Time (PhysicalDisk _Total) | localhost | 7.663 | 0.054 | 1.555 | 2.188 | 4.960 | |
| 1E-6 | Pool Nonpaged Bytes (Memory) | localhost | 10436608.000 | 10293248.000 | 10301830.100 | 30439.929 | 10293248.000 | |
| 1 | Pages/sec (Memory) | localhost | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | |
| 0.1 | Interrupts/sec (Processor _Total) | localhost | 116.341 | 66.624 | 84.636 | 18.331 | 106.005 | |
| 0.1 | Threads (Objects) | localhost | 358.000 | 357.000 | 357.714 | 0.452 | 358.000 | |
| 1E-7 | Private Bytes (Process _Total) | localhost | 264130560.000 | 263290880.000 | 263546977.500 | 353929.478 | 264052736.000 | |

从图中可以看出，CPU 使用率 (Processor Time (Processor _Total))、CPU (Processor Queue Length (System)) 的队列长度 2 个指标的曲线较为平滑，2 者的平均值分别为：11.458%、2，根据本次性能测试要求的：CPU 使用率不超过 75%，内存使用为 130M。根据 Windwos 资源性能指标的解释，一般情况下，如果“Processor Queue Length (处理器队列长度)”一直超过二，则可能表示处理器堵塞，我们这里监控出来的数值是 2，而且总体上保持平衡，那么由此推断，测试服务器的 CPU 也可能是个瓶颈。

5. 4 总结性能测试结论

根据上门测试数据，我们可以确定该系统在 50 个用户在线登陆后，系统处于超负载，只能 30 个用户成功登陆，20 个登陆失败。因此该系统最大承载能力是 30 个用户同时并发，如果想要增加承压能力，建议修改网站代码，网络环境配置，数据库的容量。在最大负载情况下，CPU 系统资源成为了系统比较明显的瓶颈，建议升级 CPU 并且提供 CPU 平台，解决对列过长问题，避免 IIS 资源成为系统软件的瓶颈。

5. 5 本章总结

本章主要是系统进行性能方面、压力方面进行测试。对系统登录时间、登录响应、负载均衡、用户并发连接等方面对主要业务模块压力和负载测试。

第六章 总结和展望

本文论述的总体框架基本上是按照两条主线展开，一个是网络通信技术和总线控制技术在楼宇自控系统中的应用；第二个是楼宇自控系统的模式设计。这两条主线相互平行，有相互影响，构成了本文的主体框架。同时，收集大量的文献和资料，并结合楼宇自动化控制系统的设计理念，充分汲取前辈和专业人事的优秀研究成果和理论事实，归纳总结，深入浅出地探讨借助网络技术和总线控制技术对楼宇自动化控制系统的应用和设计，给出合理的优化设计和节能方案，以及分析相关的一系列问题，并提出解决问题的可能。

论文中重点介绍了通过 IBMS 系统集成、统一系统通信网、统一的计算机平台构建整个建筑物的中央监控与管理界面，实现信息、资源和任务共享，以及通过可视化的、统一的图形界面，管理人员可以十分方便、快捷地对系统多包容的所有子系统进行实时监视、控制和集中的统一管理。系统结合了数据库技术、多媒体技术、Internet 技术、通信技术的应用软件系统。系统采用 C/S+B/S 方式实现。S 指数据库服务层、WEB 服务器和应用服务器；C 指报表、配置、通信管理等以传统客户端方式接入的功能模块模块，B 指用户以浏览器方式通过可视界面观察信息和数据，并向服务端发出服务请求的监控和日常管理功能模块。同时在网络技术上采用了基于 LonWorks 现场总线技术进行网络控制等。

参考文献

- [1] 杨帅,薛嵒,高安邦. 基于主机 LON 节点的研究与开发[J]自动化与仪表. 2007, 22(6): 30-34
- [2] 杨帅, 薛嵒, 高安邦. 基于 LonWorks 技术的数控机床控制系统的硬件设计[J]哈尔滨理工大学学报. 2007, 12 (2): 59-62
- [3] 杨帅, 薛嵒. 浅析基于 LonWorks 技术通信协议 LonTalk 协议[J]西安通信学院学报. 2007, 6(5): 4-7
- [4] 杨帅, 俞宁, 薛嵒, 高安邦. LonWorks 技术在智能小区中的应用[J]科学学报. 2007, (6): 32-33
- [5] 徐建俊, 杨帅, 薛嵒, 高安邦. 基于神经元芯片 MC143150 和单片机 AT89S51 的 LON 节点研究与设计[J]电测与仪表. 2007, 24(7): 57-60
- [6] 徐建俊, 杨帅, 薛嵒, 高安邦. LonWorks 技术在世界的发展的调查与研究[J]中国科技信息. 2007, (17): 131-132
- [7] 姜福祥, 杨帅, 薛嵒, 高安邦. 基于 LonWorks 技术的数字量输入节点 DI 的研究[J]科学技术与论文. 2007, 7(20): 5260-5267
- [8] 朱静, 杨帅, 薛嵒, 高安邦. LonWorks 技术通信机制的研究[J]中国新通信. 2007, (15): 35-38
- [9] 俞宁, 杨帅, 薛嵒, 高安邦. 基于神经元芯片和单片机的双处理器结构 AI 节点的研究[J]中国水运 (学术版) . 2007, (4): 139-141
- [10] 俞宁、杨帅, 薛嵒, 高安邦. 基于 LonWorks 技术的 Lontalk 通信协议的带预测的 P 一坚持 CSMA 算法的研究[J]中国科学技术学报. 2007, (7): 7-8
- [11] 俞宁、杨帅, 薛嵒, 高安邦. 基于 LonWorks 技术的数字量输出节点 DO 的研究[J]机电论文. 2007, 24(11): 8-11
- [12] 姜福祥, 杨帅, 薛嵒, 高安邦. 开放型 LonWorks 控制网络的优化[J]网络与信息, 2007, (8): 22
- [13] 徐建俊, 杨帅, 薛嵒, 高安邦. LonWorks 技术在智能大厦中的应用[J]科学咨询, 2007, (15): 33-35
- [14] 朱静, 杨帅, 薛嵒, 高安邦. LonWorks 总线控制技术在污水处理中应用[J]自动化技术与应用. 2007, (12): 23-25
- [15] 宗宏森、杨帅, 薛嵒等. 基于 LonWorks 技术双处理器结构的模拟量输出节点 AO 的研

究[J]工业控制计算机. 2007, 20(11): 7-9

- [16] 智淑亚,高安邦,杨帅. 基于 LonWorks 现代智能测控系统的开发与应用[J]电脑学习,2006,1
- [17] 徐建俊,俞宁,杨帅等. 基于 LON 现场总线技术的电力线收发器 PLT-22 的设计[J]电力自动化设备.2008, 28(9): 108-111。
- [18] 杨帅. 基于神经元芯片和单片机双处理器结构的 LON 节点的研究[D] 哈尔滨理工大学.硕士学位论文.2007。
- [19] 薛嵒, 杨帅, 史宜巧等.Pyxos 嵌入式网络中链路电源模块的设计[J]电子工业专用备, 2008(11): 43-47。
- [20] 张建玲. 楼宇自动化系统建设原则及参考方案。《湖南电力》 2000 第 6 期
- [21] 何志议. 智能建筑中的楼宇自动化系统理论浅析。《电工技术杂志》 2004 第 8 期.
- [22] 建筑物自动化系统技术综述-Excel5000 设计手册。2000
- [23] 张小军. 基于 LonWorks 技术的楼宇自动化系统的研究[D]哈尔滨:哈尔滨理工大学.2003.
- [24] 张守勇. 基于 LonWorks 技术的智能小区管理系统[J]电子技术杂志.2004(3):84-88
- [25] Jin W.Stewart G.Culshaw B.Absorption measurement of methane signal processing[J]Opt.Lett.1993.18(16)I 1 363-1 366
- [26] YE Xian-feng. TANG Wei-ZHONG Application of near infrared laser diodes to gas sensing[J]Semiconductor Optoelectronics.2000.21(2).
135—138(in Chinese)
- [27] 赵新民等. 智能仪器设计基础[M]哈尔滨:哈尔滨工业大学出版社.1999
- [28] 张友德等. 单片微型机原理、应用与实验[M]上海:复旦大学出版社.2002
- [29] 马腾等. 基于 CAN 总线可燃气体探测系统设计[J]传感器世界.2006.12(11)-34-37
- [30] 方佩敏. 新编传感器原理、应用与电路详解[M]北京:电子工业出版社.1994
- [31] 陈小忠等. 单片机接口技术实用子程序[M]北京:人民邮电出版社.2005
- [32] 翁晓霞. 《开发基于 WEB 的远程教学系统的关键技术比较》 [J]计算机工程与应用, 2000,
- [33] 汪云. 《基于 B/S 模式的学生信息管理系统》 [J]襄樊学院学报.2006 (7)
- [34] 许丽丽. 《AJAX 技术在成教教务管理系统中的应用》 [J]计算机时代.2008 (5)
- [35] 陈炎, 杨庚. 《基于 Web Services 的电子钱包系统的分布式解决方案研究》 [M]南京邮电学院学报, 2005, (1): 42-45.
- [36] 郑 刚. 《基于 B/S 模式的网上教学系统研究与设计》 [J]微型电脑应用.2001 (5)
- [37] 贾 强,张园林. 基于 B/S 模式的网络辅助教学系统的研究与设计[J];电脑学习;2003 年 06 期
- [38] 邓艳平,方丽菁. 网上教学体系结构和安全性分析[J];广西民族学院学报(自然科学版);2002 年 S1 期

- [39] 李健.基于 Internet 远程教学系统中应用体系结构的研究[J];湖南广播电视台大学学
- [40] 何克抗, 李文光.教育技术学[M]北京: 北京师范大学出版社, 2002.11-15.
- [41] 辛希孟.信息技术与信息服务国际研讨会论文集: A 集[C]北京: 中国社会科学出版社, 1994.20-33.
- [42] 张铮, 何秀玲, 冯刚等.基于 WEB 教学素材库的模型设计[J]计算机工程与应用, 2001(11): 112-132.
- [43] 张翼飞, 李胜宇, 徐蕾. 数据库连接池技术在 JSP 数据库访问中的应用[J] 微处理机, 2005,(01)
- [44] 谷震离. ADO 访问 SQL Server 数据库技术分析及其应用[J] 计算机应用与软件, 2004,(12)
- [45] 余健, 窦丽华, 陈杰; 基于 SNMP 协议的网络主机综合监控方法研究[J];北京理工大学学报;2002 年 03 期
- [46] 刘桂山, 胡军程; 1553B 总线信息流设计[J];北京理工大学学报;2003 年 03 期
- [47] 蓝波, 杨琴, 潘渊颖; 基于 SNMP 的 MIB 库访问分析及接口的实现[J];北京石油化工学院学报;2003 年 03 期
- [48] 许波, 郑少仁; 远程教育平台管理控制系统的设计与实现[J];北京邮电大学学报;2003 年 S1 期
- [49] 古玉洁, 吴宇红, 刘炯; 基于影子代理的战术网网络管理系统的研究[J];北京邮电大学学报;2004 年 S2 期
- [50] 谭跃生, 王静宇; 管理信息库(MIB)的分析及其在网络计费中的应用[J];包头钢铁学院学报;2001 年 02 期
- [51] K.Pearson.Exploring process data,Journal of Process Control,2001,(6):179-194
- [52] Enterprise Buildings Integrator Lonworks Interface Reference.Honeywell Company,2003.
- [53] CHRISTIANSSONP. Knowledge representationsand informationflowin the intelligent building[C] //Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering. VirginiaUSA: [s. n.], 2000:1-8.
- [54] BONDEM. Intelligent Building Installations with Lon-Works[R] Copenhagen: LonUserGroupDenmark, 1999.
- [55] LonworksProtocol[S] CAUSA: EchelonCorp., 1993.
- [56] 刘琼发 刘寿强. 楼宇自动化系统设计方案。《现代电子技术》 2003 第 3 期
- [57] 王松青 童明淑.LonWorks 与 BACnet 现场总线技术及其在楼宇自动化系统中的应用 《低压电器》 2004 第 9 期
- [58] 赵起升 朱静孙 王平. 智能建筑中的楼宇自动化设计及其应用. 《华中科技大学学报: 城

市科学版》 2003 第 3 期

- [59] Echelon Version Co. Introduction to the LonworksSystem[R]. Echelon Corporation.1.0,2001: 1-23, 45-47, 101-145, 144-176
- [60] Echelon Corporation, Echelon 公司消息报道[R], www.echelon.com.cn.