

密级：

论文编号：

中国农业科学院

学位论文

商品猪精细养殖生产管理数字化平台
的构建与实现

**The Construction and Realization of the Production
Management Digitization Platform for Precision Feeding of
Commercial Pig**

硕 士 研 究 生： 苏 希

指 导 教 师： 刘世洪

申请学位类别： 管理学硕士

专 业： 管理科学与工程

研 究 方 向： 计算机农业应用

培 养 单 位： 农业信息研究所

提交日期 2005 年 6 月

Master Thesis

The Construction and Realization of the Production Management Digitization Platform for Precision Feeding of Commercial Pig

Master Candidate: Su Xi

Supervisor: Liu Shihong

Graduate School of CAAS
Institute of Agricultural Information, CAAS
Beijing 100081, PR China

June 2005

论文评阅人：

薛领

赵瑞雪

答辩委员会主席：

胡伯海

答辩委员会成员：

李锦涛

王文生

诸叶平

薛领

答辩时间：

2005-6-28 上午 9 时

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了本文特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国农业科学院或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：

时间：

关于论文使用授权的声明

本人完全了解中国农业科学院有关保留、使用学位论文的规定，即：中国农业科学院有权保留送交论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意中国农业科学院可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

研究生签名：

时间：

导师签名：

时间：

摘要

现代企业广泛采用数字化技术，通过信息资源的开发和利用，不断提高其生产、经营、管理、决策的水平和核心竞争力。相比之下，我国养殖企业普遍存在生产管理落后、信息不畅、投资市场风险大等问题。本研究针对这些问题，以商品猪养殖企业为例，提出利用计算机技术构建商品猪精细养殖生产管理数字化平台，对所涉及的对象和全过程进行数字化的表达、设计、控制和管理，把信息技术作为农业生产力的重要要素，应用到商品猪养殖的各个生产环节并使之成为不可缺少的组成部分。同时，由于养殖企业具有集约化、生产规模化、管理规范化的特点，具备了应用数字化技术的可行性。

本研究的核心内容是商品猪生产管理和饲料管理的数字化平台构建。在充分调研分析基础上，运用面向对象的分析和设计技术，进行平台的需求分析和功能设计，并采用 Visio 等建模工具完成了系统设计。在对面向对象技术和结构化技术、J2EE 和 .NET 两大框架技术、C/S 和 B/S 开发模式、常用的 Web 开发语言、常用的 Web 数据库等技术综合对比分析的基础上，决定采用 MVC 的设计模式并结合组件技术的方案。生产管理部分实现了猪只进场记录管理、生产信息管理、饲料投放管理和销售管理等功能；饲料管理部分实现了原料管理、配方管理、库存管理和配方计算等功能。此外，采用数据库优化技术、Web 服务器优化技术、Session 技术、安全策略等，对系统的性能和安全性进行了设计。为加强系统实用性，以山西宏明养殖场为基地，将系统投入实际运行，积累用户反馈。

通过系统的实现和实际应用得出以下结论：（1）综合运用数字化技术对商品猪的生产过程进行全方位管理是可行的、有效的。（2）采用面向对象的分析和设计方法，以商品猪对象为中心，对实际生产进行流程模拟和行为模拟，便于构造出模块化、可重用的系统。（3）采用开放性好的 B/S 体系结构，在伸缩性、可扩展性以及安全性方面具有更大的优势。（4）采用 MVC 设计模式，使得应用程序的结构更加清晰，层次间关系明确，增强代码稳定性。（5）采用组件技术，极大的扩展程序的功能，提高了开发的效率和质量，便于维护。

关键词：数字化平台，商品猪，精细养殖，组件，MVC

Abstract

Digitization technology is widely used in modern enterprise to improve the level of its production, management, administration, decision-making and core capability of competition. Nowadays, most of the domestic feeding enterprises have such problems as uncultured management, bad-informed information and large venture of investment. A production management digitization platform for precision feeding of commercial pig is proposed under the background. By computer technology, which digitizes the expression, design, control and management for the object and the whole process. The information technology being looked on as important factor of agricultural productivity is utilized to each courses of production in commercial pig feeding which causes them to become indispensable content. At the same time, the characters of intensivism, produce scale and management standardization the feeding enterprises own increase the feasibility of digitization technology.

The core of the study is the set-up of the digitization platform for production management and feed management for commercial pig. On the basis of abundant investigation and analysis, the demand analysis and the functional design of the platform have been realized through object-oriented analysis and design technology. The system design has also been completed with the modeling tool such as Visio and etc.. Through synthesis contrasting and analysing the object-oriented technology and the structuring technology, frame technologies as J2EE and .NET, development pattern like C/S and B/S, commonly used Web script language as well as some commonly used Web database, the scheme using design pattern of MVC combined with component technology is brought forward. The function of pigs-coming record management, production information management, feed put-in management and so on have been realized in the production management part in the platform; and the function of raw material management, formula management, stock management and formula calculation and so on in feed management part. Moreover, the design for system performance and security have been completed by database optimization, Web serve optimization, Session technology and security tactic. In order to strengthen practicability of the system, has been carried out in contacted by myself, and the platform was put into spot inspection to accumulate users' feedback with the example of Shanxi Hongming breeding farm.

Through the realization of the system and the application, the following conclusion have been summed up: (1) It's feasible and effective to manage production process comprehensively by the synthetical exertion of digitization technology. (2) It's convenient to form modularized, reusable system by simulate the flow and deed of production with the center of commercial pig as the object and through the method of object-oriented analysis and design. (3) The well-open B/S structure is more advantageous for retractility, extensibility and security. (4) MVC design model increases the stability of the code, and makes the structure of the application more clear, the relationship of hierarchy more specific. (5) Component technology enlarges the function of the program, and improves the efficiency and quality for development and it's convenient to maintain the system.

Key word: Digitization Platform, Commercial Pig, Precision Feeding, Component, MVC

目 录

第一章 前 言	1
1.1 问题的提出.....	1
1.1.1 农业企业数字化技术的广泛应用.....	1
1.1.2 畜禽养殖企业需要数字化技术.....	3
1.1.3 商品猪精细养殖生产管理数字化平台的提出.....	4
1.2 国内外相关领域概况	5
1.2.1 畜禽生产管理数字化.....	5
1.2.2 畜禽育种数字化	5
1.2.3 个体动物营养需要模型与日粮优化系统数字化.....	6
1.2.4 畜禽疫病诊断数字化.....	7
1.2.5 饲料与畜产品安全监控数字化.....	7
1.2.6 畜禽精细养殖数字化平台研究进展.....	8
第二章 系统设计的技术理论基础.....	10
2.1 面向对象的分析与设计技术.....	10
2.1.1 结构化方法	10
2.1.2 面向对象的方法	10
2.1.3 面向对象方法与结构化方法的比较.....	12
2.2 组件技术.....	13
2.2.1 J2EE 技术框架介绍	13
2.2.2 .NET 技术框架介绍.....	17
2.2.3 两大技术体系比较.....	19
2.3 web 开发技术	20
2.3.1 C/S 模式 、B/S 模式和混合模式.....	20
2.3.2 目前流行的 web 开发语言介绍	24
2.3.3 Java 语言介绍	27
2.4 web 数据库技术	30
2.4.1 web 数据库介绍.....	30
2.4.2 常用几种 Web 数据库比较	31
2.4.3 Web 数据库的安全性	32
2.5 报表技术.....	34
第三章 系统分析与技术路线.....	36
3.1 研究目标.....	36
3.2 主要研究内容.....	36
3.2.1 生产管理	36
3.2.2 饲料管理	40
3.2.3 生产监测	42

3.3 研究方法.....	42
3.3.1 技术路线	42
3.3.2 采用的系统架构	43
第四章 系统的设计与实现	45
4.1 系统总体设计	45
4.2 系统功能模块设计	46
4.2.1 总体功能模块划分.....	46
4.2.2 所参与的子模块功能详述.....	47
4.3 数据库结构设计	57
4.3.1 数据库设计的原则.....	57
4.3.2 主要表结构	58
4.4 系统开发平台的建立	61
4.5 采用的关键技术.....	61
4.5.1 数据库优化技术	61
4.5.2 Web 技术	62
4.5.3 MVC 模式的应用	63
4.5.4 组件技术应用	64
4.5.5 其他关键代码举例.....	68
4.6 系统安全性设计	70
4.6.1 操作系统的安全	70
4.6.2 SQLServer 数据库服务器的安全.....	70
4.6.3 应用系统的安全	71
4.7 系统实现	72
4.7.1 系统运行软件环境及配置.....	72
4.7.2 系统运行	74
第五章 系统应用案例	78
5.1 工厂化养猪的条件和优势.....	78
5.2 山西宏明养殖有限公司简介.....	79
5.3 实施效果	79
第六章 总结与讨论	81
参考文献.....	83
致 谢.....	86
作者简历.....	87

第一章 前言

1.1 问题的提出

1.1.1 农业企业数字化技术的广泛应用

现代企业利用现代数字化技术,通过信息资源的开发和利用,不断提高生产、经营、管理、决策的效率和水平,进而提高企业经营效益和企业竞争力。这个过程的核心就是降低成本和改善管理水平。企业数字化技术应用是一个很广泛的概念,具体可以分为三个层次。第一层是企业实现生产自动化,如生产设计自动化(CAD)、自动化控制等。第二层是企业数据的自动化、信息化,用数字化技术对生产、销售、财务等数据进行处理。第三层是更高层次的辅助管理、辅助决策系统 MRPII、CIMS、OA 等(唐鸿儒,2003)。

农业企业作为我国农业生产的主力军,它与农村经济相比,具有人才、科技、管理和经济等多方面的优势。另外它延续了计划经济合理的成份,如工、农、商、建、服等产业合理布局、宏观调控等(刘世洪,2003)。因此,农业企业的数字化具有重要的现实和可操作性,得到了广泛的应用。

1.1.1.1 农业生产管理系统

数字化技术给农业管理带来了高效率、高质量和高效益。用计算机辅助农业计划管理可以增加产值,缩减管理费用,减少消耗和废品率,管理种养业生产可增加收入。上海申星奶牛场自 1995 年起将计算机技术应用于奶牛生产管理,从奶牛生产数据收集、处理,到信息反馈已初步形成牧场管理信息化,将奶牛生产纳入了有序的科学管理轨道。此外,农业部开发和建设了乡镇企业管理信息系统,农业经营管理信息系统,农村能源及环境监测管理信息系统。已见报道的国内农业管理信息系统还有“全国森林病虫害防治管理信息系统”、“国内植物检疫管理信息系统”、“林火管理信息系统”等,各省市也开发了各自的农业管理信息系统等。这些农业信息系统的开发和建设,使整个农业行业的管理逐步走上了自动化、科学化的轨道。

1.1.1.2 农业决策支持系统

农业决策支持系统以数据库、模型库、知识库和推理机为基础。知识库包括领域专家用来解决生产问题的定性、定量知识;推理机控制整个程序的运行,根据知识库进行综合推理和决策。建立知识库与推理机以指导农业生产的管理决策,具有较强的实用性和决策能力。将模拟模型的动态预测功能和专家系统的推理决策功能相结合,使其对农业生产系统具有较强的解释性和预测性。中国农业科学院农业信息研究所开发的“农业企业综合生产能力辅助决策支持系统”,面向农业企业生产经营管理者,运用现代网络技术、Web 服务技术,建立适合我国国情的农业企业综合生产能力分析系统,实现农业企业的生产管理决策信息化、网络化,提高企业宏观决策水平。系统包括农业企业综合生产能力投入水平

现状分析、农业企业综合生产能力投入水平预测分析、农业企业综合生产能力产出能力分析和农业企业综合生产能力投入要素需求预测分析。

1.1.1.3 农业生产控制系统

数字化技术使农事操作实现了自动化,信息化与精确化。农业自动化的监控系统,是将环境监测、数据采集、数据分析、数据传送与环境控制的软件和设施相结合的整套系统。

目前在温室与大棚的自动化控制方面已经相当成功。根据天气与环境的变化,根据作物的需要的变化而进行调控。因此,不管天气与环境怎么变化,都可以培育出高质量的、无污染的、规格化的农产品。

在田间作物的灌溉方面,根据天气,土壤水分与作物需求的变化,控制灌溉量、灌溉时间的自动灌溉系统,有限水域(如池塘)的水产环境的自动监控系统,畜禽饲养的自动监控系统,国内外都已经有了成功经验。

在粮食、果品、蔬菜、花卉、畜禽产品的保鲜、储藏的设施中,自动化的技术也都很成功。至于微生物的发酵工程,农业菌藻生物的培育工程与农产品加工工程等方面,完全可以应用信息技术,实现全面的自动化。

1.1.1.4 农业监测系统

地理信息系统(GIS)与遥感技术(RS)是信息技术应用于农业环境资源监测的主要手段。地理信息系统是近20年来国际地理科学的一项突破性的技术。它将系统科学、信息科学、计算机的数据采集、处理和分析模型,数据库技术与计算机图像技术密切结合起来,成为一种综合性技术。

地理信息系统与遥感技术的结合应用,使农业资源环境的研究与监测方法得到根本性的改观。其应用领域主要有:农业土地、耕地、土壤、森林、草原、水面等各种农业资源的探测、评价与动态监测;各种农业灾害(洪涝、干旱、风暴、病虫害等)的实时预测与监测;各种农业作物面积与产量的预测与监测;农业环境(大气、土壤等)污染的预测与监测;各种农业与园艺作物、畜禽水产生物、经济林木的地区适应性分布的研究,可为农业产业结构调整提供依据。

1.1.1.5 农产品电子商务系统

电子商务(E-commerce)始于90年代,是指利用电子网络进行的商务活动。目前关于E-commerce市场前景的预测每过3个月就要更新一次,前景光明(李琪,1997)。农产品电子商务技术加快市场信息化建设步伐,形成反应灵敏、覆盖面广的市场信息网络,是农产品电子商务技术的用武之地。电子商务有四个发展阶段:第一阶段侧重于广告宣传;第二阶段实现网上交易和结算;第三阶段发展到数据交换(EDI);第四个阶段实现真正意义上的网络全球化。我国电子商务大多处于第一和第二阶段,农产品电子商务技术的研究和开发应用潜力巨大,其重点是有效的实现农产品和农业科技信息的网上交易,研发高效、安全的查询、订购、支付技术,以及交易规则、管理制度、技术手段和配套设施等。对传统农产品批发市场与电子商务进行整合,将批发市场变成“农产品市场数码港”,从而有效地

连接生产与销售终端，减少中转消耗，加速物流运转速度，使交易成本降至最低，市场半径辐射全国乃至世界。

1.1.1.2 畜禽养殖企业需要数字化技术

二战后殖民制度瓦解，工业化国家被迫改造自己的传统农业，从依赖扩张土地、进口粮食，改为依赖科学技术，提高生产力，依靠“垂直扩展”，提高生产效率。商品农业是以市场需求为中心，以盈利为目标，国家的调控只能靠政策、税收、利率等间接手段。直接指挥农业生产的是利润（刘少伯,2003）。现代化的商品农业挽救了他们的经济发展，使发达国家由依赖殖民地进口农产品转变成自给有余，以至成为农产品出口国。相反的，传统农业国许多都依赖于工业国。看来，农业发展要依靠科技，特别是信息科技。

农民目前迫切需要解决的是信息和决策功能。现代商品农业，产量是提高了，但也增加了农民的困惑，首先面临的是风险越来越大，增加农业收入越来越困难，比重越来越低。国家处理农业中的问题也越来越多。

对于畜禽养殖企业来说，由于其特点自然而然的产生了一些问题。

1.1.1.2.1 管理难度加大

养殖业本身的发展趋势是生产规模越来越大，集约化程度越来越高，因而管理的难度也越来越大，仅仅凭经验已是远远不够了。养殖场正面临着由基于经验的生产系统到基于信息和科技的生产系统的转变，这无疑需要信息技术的支持。

1.1.1.2.2 市场和投资方向的不定性

农产品的需求和供应之间，始终处于周期性波动之中，而且农业投资的生产要素肥料、水力、设施、种畜、机械等投入越来越多，这些现象常常令农民难以琢磨，提心吊胆，对农产品市场，伤透了脑筋。对于畜禽商品，经常遇到提“买难”和“卖难”的问题。在信息不畅，缺乏获取渠道的情况下，曾出现过宰杀母猪、烫死雏鸡的现象。农民需要通过某种渠道，去及时获取市场和投资信息，来指导农业生产。

1.1.1.2.3 缺乏实用技术

技术也是个“无底洞”，永远需要更新，以新开发的“新技术”，如品种、疫苗、设施、管理替代现有技术。家畜会抗药，边际效益会降低。缺乏更新技术的机制，缺少获取信息的渠道，农民无法掌握最新最快的国内外实用技术，也就无法扩大生产规模，增产增收。

1.1.1.2.4 生态成本增加

家畜粪便污染，草原生态破坏，最终是要农民支付生态成本，成为不堪忍受的重负。现在上海、北京、广州等大城市也面临这个问题。

生产者、经营者和领导者总是处于十分困难的决策之中。人们经常苦恼的是：到底生产什么？生产多少？谁是我们的上帝？怎样对付十分复杂的商品生产、交换、分配和消费。现在，非能力不足，而是脑力不够，不是没想到，而是想不出办法（刘少伯,2001）。

长期以来,人们为了摆脱市场危机总是在千方百计地寻求解决办法,最后总结出经验,就是依靠“数字化技术”,去管理控制生产过程,改进生产工艺;及时获取市场和投资信息,指导投资方向;利用计算机合理调配资源,避免浪费,保护环境,实现可持续发展。

1.1.3 商品猪精细养殖生产管理数字化平台的提出

日本经济学家竹中一雄提出:“农业是最费脑筋、难度最大的产业,工业的基础是数理化,农业的基础广泛门类繁多,除数理化外,还有生物、气象、经济……,世界发达国家都是先解决工业化,然后才解决农业问题”。人的大脑实际是通过四项功能实现的,体现了思维、决策,及付诸行动。即:信息的感测收集;将信息传送到大脑;对信息的加工、优化,即思考;输出信息,付诸行动。人类不能制造机器大脑,但是可通过放大四项功能,体现强化智能,增加智慧,提高决策能力,对付复杂的商品农业。主要任务有:

1.1.3.1 信息的采集

广泛采用现代信息技术,对畜禽养殖生产过程中进行全方位监测和控制,获取各种实时性的生产数据,保证数据的真实性和有效性。美国有句名言:“输入的是垃圾,输出的也必然是垃圾”。这里指的是你制造“假数据”给计算机,它也给你造个“假结论”。可见,信息的采集是基本功(甘仞初,2000)。畜牧行业需要下大工夫,建设真实的数据信息采集系统。

1.1.3.2 信息的传输

网络信息共享和电子商务,是经济发展的前沿。我国信息产业的基本建设发展很快,不断扩大的卫星、光缆通讯,使实现信息共享越来越容易,进入经商、教学、购物是指日可待。目前交换信件、检索信息、从事商业、科技活动已具备雏形。当然,首先就是交换信件、信息检索。进入网上的商务活动还需要有个过程。而网上商务活动恰恰是最具有活力,最能使网络存在、壮大的经济基础。

1.1.3.3 信息的优化

开发畜牧软件系统。强化智能,增加智慧,提高决策能力的关键是开发性能高、功能强的多种软件。提高畜牧技术、加强企业管理,增加农民收入,提高转化效率,降低生产成本都要依靠软件。即将收集的数据经计算机模拟计算结果,根据计算的利润、成本、资金、销售、工资等参考结论,做出管理决策。

1.1.3.4 信息的控制

前面三项都是参谋作用,提出方案。而发号施令,付诸行动,避开农业风险,增加农民收入,增强行业竞争力,还需要依靠行业或行政领导的指挥艺术。现在国外大企业都有“知识主管”,专门从采集的大量信息中筛选、提炼,制定可供选择的方案,为领导指挥决策服务。

针对我国养殖企业目前普遍存在生产管理落后、信息不畅、投资市场风险大等问题,以商品猪养殖企业为例,可以通过建立一个集信息收集、信息传输、信息优化和信息控制

为一体的技术平台,对商品猪养殖企业所涉及的对象和全过程进行数字化和可视化的表达、设计、控制和管理,运用计算机对农业进行辅助决策,增强大脑的功能,把信息技术作为农业生产力的重要要素,应用到养殖的各个环节并使之成为不可缺少的组成部分。

1.2 国内外相关领域概况

1.2.1 畜禽生产管理数字化

Cross (1990) 开发了供爱尔兰牛场用的计算机化管理信息系统 ERIRYMZS, 加拿大 QventinME. 设计了专用猪粪肥管理的 GLS 模型系统,英国的 Qvigley 为奶牛场研制了计算机化管理记录与检索系统, Hlirne R. B. M. 开发了用于猪场业绩的综合决策支持系统和专家系统。

国内在这方面也进行了一些研究。朱益民 (1983) 使用 DBASE 语言研制了奶牛信息专用数据库。陈德作 (1991) 建立了奶牛生产信息电脑管理系统。该系统包括数据库管理、信息查询、日常管理、报表打印、饲料配方、育种值估计、统计分析和产奶量预测等功能;邢广亚 (1992) 利用 Foxbase 研制了蛋鸡场生产管理数据库系统,包括建立数据库、系统维护、信息查询、打印报表、绘制图件和咨询等功能;刘少伯 (1992) 利用 Foxbase 研制了蛋鸡场经营管理专家诊断系统;金福昌 (1993) 研制了大规模种禽场计算机辅助生产管理系统,该系统集计算机技术、应用数学、畜牧业管理技术于一体,在生产中应用产生了良好的经济效益和社会效益;王林云 (1991) 研制了猪育种生产微机系统 SBP,包括种猪数据库、数理统计、饲料配方和配种预产等子系统;广东农业机械研究所开发了工厂化猪场计算机管理信息系统 PPMIS;亚卫畜牧新技术有限公司开发了现代化养猪生产管理及分析系统 PigMap,该系统按照工厂化养猪全过程五段式设计,使用 Visual Foxpro 编程,在 Windows 操作系统下运行,整个系统包括生产管理、育种管理、饲料配方、疾病诊断、财务管理、仓库管理、人事管理和客户管理等八个相互独立的子系统,有基础数据的输入与维护、计算与分析、综合查询、打印统计报表等部分,具有读、增、删、改、统计运算、查询和决策等功能,“九五”期间已在攻关示范场中应用 (PigMap 产品介绍,2005)。陆昌华 (1991) 研制了蛋鸡生产管理电脑辅助管理系统;李振卿等开发了鸡群结构动态仿真软件;吕胜利等也研制了畜群发展的系统动态仿真模型,在对甘南草原绵羊发展规划的论证中得到了具体应用。

1.2.2 畜禽育种数字化

在猪的育种中已广泛应用 BLUP(最佳线性无偏预测)方法计算育种值。近年来由于计算机技术的发展和生物技术在动物育种中的应用, BLUP 育种值估计方法又有所发展,如从公畜模型发展为动物模型、单性状育种值估计发展为多性状育种值估计、常规繁育体系的育种值估计发展为有胚胎移植、胚胎切割等非常规繁育体系的育种值估计等 (全国种猪遗传评估方案, 2003)。

奶牛体型线性评分电脑化：奶牛外貌体型线性评分是奶牛育种工作的一项重要内容，在发达国家得到普遍重视。近年来，我国也开展了此项工作，并已先后推广到北京、天津、武汉、内蒙古、山西、辽宁、河北等地。由于此法评定奶牛项目繁多，数据处理工作量大，中国农业大学师守堃等经过长期研究，完成了奶牛线性评定信息处理分析系统系列软件的研制。黑龙江省家畜指导站葛星明等，也研制了奶牛线性外貌评定的系列软件，不仅可计算出部位及整体线性评分，而且还可计算公牛各性状、部位及整体评分的重复力(Rep)、遗传传递力(ETA)、标准化传递力(STA)等，还可打印出各公牛的线性外貌柱型图，从而使评定工作实现了电脑化。

在肉牛育种研究方面，中国农业科学院畜牧研究所 20 年来，分别在新疆、四川、内蒙古、吉林等省(区)的 7 个国家级种畜场和 5 个育种基地、进行开放核心育种体系(ONBS)多血缘系的核心群选育，建立起开放核心育种体系(ONBS)，西门达尔牛种群规模达 100 万头，为进行 ONBS 育种智能系统的研究提供平台。

国际上，美国、加拿大、日本、澳大利亚、新西兰和主要的欧盟国家，甚至在发展中国家印度，在他们的家畜、家禽国家育种中心，基本上都通过网络通讯技术在全国范围内建立优良种畜、种禽的系谱数据库，并通过软件支持联合育种(家畜)和多性状的 BLUP 法育种。

尽管国内外从不同的研究及应用层面开发了大量的育种应用软件，但由于缺乏标准的数据接口与数据标准，这些功能相对单一的系统不能满足构建精细养殖技术平台的需要。因此，需从构件软件设计的技术出发，尽可能开发具有软件构件特性的各类育种智能系统。

1.2.3 个体动物营养需要模型与日粮优化系统数字化

肉牛营养需要量的预测模型研究方面，主要成果基本集成在 NRC《肉牛营养需要》(1996)第 7 次修订版的内容中。该版本主要考虑了动物品种、生产水平、体况、环境与应激对采食量的影响等因素，通过定义两种模型层次计算肉牛的营养需要量。第一模型层次属于经验型，类似过去用表格列出营养需要量，强调预测营养需要量的准确性，可直接用于指导生产；第二模型层次则依赖建立的机制模型，在事先定义某些最基本的有关养分消化的参数数据或关系后，通过驱动预测模型计算肉牛营养需要量。它是以美国康奈尔大学在上个世纪 90 年代提出并建立的碳水化合物和净蛋白质体系(CNCPS)为基础而建立的一系列预测模型。

与 NRC 第 7 次修订版《肉牛营养需要》相比，其《商品牛营养需要》(NRC,2001)全面系统地采用了计算机数学模型描述动物处在不同环境、不同生理阶段的营养需要量，同时提供的数学模型能够将不同饲料原料对营养物质吸收的差异和对动物生产性。

能造成的潜在影响考虑在内，从而可以动态调控饲料养分的供给，并达到预测的生产目标。《奶牛营养需要》(NRC,2001)也提供一些营养需要量表，但表中数据都是通过模型计算列出的，并非经验性数据，同时提供了计算机模型程序预测商品牛营养需要量。所以，与《肉牛营养需要》的研究比较，其模型化研究又深入一步（孟庆翔等译,2004）。

在猪的营养需要量的模型研究方面, NRC(1998)第十版《猪营养需要》, 系统全面提出了预测猪对各种养分的需要量数学模型, 为实现猪个体营养调控和通过计算机模拟猪的生产与养分的供给提供了可能(譙仕彦译,1998)。

在国内, 结合近 14 年的研究与饲养实践及国际研究动态, 由农业部畜牧兽医局列项的, 由中国农业科学院畜牧研究所、中国农业大学、内蒙古畜牧科学院等单位承担的《奶牛饲养标准》、《肉牛饲养标准》和《肉用绵羊和山羊饲养标准》的制定工作已经完成, 为进行我国反刍动物个体营养需要量的科学预测与日粮的精准配制提供最直接的模型、数据及营养体系。

在配方技术及数据资源方面, 上述单位, 经过三代人的不懈努力, 开发完成了我国第 1~7 代饲料配方系统; 并在动物营养参数积累上, 一直维护《中国饲料数据库》的资源建设, 为进行猪饲料矿物元素的 GIS 平台、畜产品及饲料质量的网络数据库建设奠定了必备的数据资源。

在畜禽营养与饲养领域研究总的趋势是, 通过数学模型, 将影响动物个体养分需求的诸如品种、环境、生长模型和饲料特性等因素考虑在内, 动态精准预测营养需要, 并进行日粮的优化, 达到精细饲养的目的。

1.2.4 畜禽疫病诊断数字化

日本东京农大杉木等以大型养鸡场积累的资料为基础, 研究蛋鸡产蛋生产预测模拟模型, 用于诊断疾病。九十年代, 美国开始利用笔记本电脑, 到农场巡回医疗, 给家禽看病。

我国学者许剑琴教授主持并完成了国家自然科学基金项目“兽医中西药处方生成系统的研究”, 此项目的研究结果为计算机在畜禽疾病诊疗方面的应用奠定了基础。江苏省农科院农业现代化所研制的“鸡常见疾病临床诊断专家系统”, 不但采用了用户二次加权的方式来提高诊断符合率, 而且引入了人工神经网络技术来改进知识获取方法, 进一步提高了诊断符合率, 达到了临床诊断专家的水平(陆昌华等,1999)。系统可对鸡新城疫、马立克、法氏囊、支气管炎、喉气管炎、霉形体病等主要传染病、鸡营养代谢病和鸡寄生虫病等做出各种诊断, 给出疾病的防治措施、防治计划与预后, 并输出进一步诊断, 及实验室工作操作程序。系统曾对 244 份病例验证, 结果表明: 完全符合 197 份; 完全符合率达: 80.74%; 基本符合 21 份; 基本符合率: 8.61%, 达到实用水平。

1.2.5 饲料与畜产品安全监控数字化

在欧洲, 已经建立了“从农场到餐桌”的全过程质量控制模式, 自“疯牛病”和“二恶英”事件在欧洲爆发后, 欧盟国家消费者对畜产品产生信心危机和遭受巨大经济损失, 在总结了食品安全管理的经验教训, 开始实施这种模式(赵昕红,2001)。其基本原则就是针对畜产品产出过程中, 养殖、屠宰、加工、运输和贮存(上市)的每个生产环节均实行相应的质量保证体系, 来确保其生产过程的安全性, 从而最终满足消费者获得安全畜产品的需求。其主要控制点有: ①农场安全卫生跟踪系统。在畜牧业发达的欧洲国家, 政府已出台相应的法规, 要求养殖场必须建立“安全卫生计算机跟踪系统”(王立方等,2004)。该系

统记录包括了动物的标记编号、合同兽医来访次数及每次的用药情况、饲料公司的名称、运输动物的公司及所用的运输工具和动物所去屠宰厂的名称。同时还记录与农场环境卫生有关的关键数据；②饲料生产的安全控制。包括：通过立法和强制性标准来规范安全饲料的生产过程。迄今为止，欧盟共颁布了与饲料生产和动物饲养有关的法规和强制性标准近 100 项，其涵盖的内容有采样和分析方法；饲料产品和饲料原料安全评价；兽药使用、兽药和农药残留限量、营养物质分类；毒素和有害物质的控制；禁用物质名录；动物保健与营养；添加剂使用与使用条件；酶制剂和微生物制剂的使用；为特殊营养目的使用的物质名录；饲料原料的循环使用，饲料的销售、运输和标签；③通过将良好动物营养操作规范 (AGMP) 与 ISO9001 或 ISO9002 认证相融合用于饲料生产的安全管理全过程；④畜产品加工实施危害分析及关键点控制(HACCP)、实施畜产品标签立法。要求畜产品标签上必须标明产品名称、质量等级、贮存方式和保质期(农业部畜禽产品质量监督检验测试中心,2003)。

自从日本发现 BSE(牛绵状脑病)之后，随着民众对健康的关注，日本连锁超市 Aeon 在其门店中已经安装了牛肉追踪计算机系统，告诉消费者关于正在销售的牛肉信息。通过输入特殊的代码，该系统允许客户追踪指定盒包装的牛肉，输出该块牛肉生产的全部信息，包括 BSE 检测、饲料、品种以及其他信息。

美国在建立红色肉类产品追溯系统方面滞后于其竞争对手如欧盟、加拿大、澳大利亚和新西兰，而我国尚未建立有效的畜产品安全跟踪体系和实用的计算机信息系统（乔娟,2002）。

1.2.6 畜禽精细养殖数字化平台研究进展

为支撑精细养殖而必须的养殖信息自动获取和智能管理控制系统的研究和应用方面，在国内外已取得了一些研究进展，为本研究获得的精细养殖技术平台应用奠定了一定的基础。

在七十年代后期，欧美发达国家的商品牛场管理信息系统中已得到成功应用。基于奶牛编号自动识别器的个体产奶量定量配料和饲养管理系统，自八十年代初已开始在欧美各国农场中推广使用。奶牛个体编号电子自动识别器作为自动鉴别动物个体编号的基础，于七十年代中期即研制成功商品化产品（苏建华等,2000）。近几年，一种养猪场推广应用的耳挂针杆式、塑料密封、重量小于 5 克的微型个体编号识别无源信号转发器，不仅可用于按个体发育和健康状况进行定量自动配料，建立动物个体管理信息档案，自动监测体重增长、母畜发情期、健康指标自动诊断等，与便携式数据采集器结合，还适用于不同规模的牲畜科学饲养管理（朱能武等,2000）。英国八十年代后期研制的平养肉鸡场鸡群体重信息采集系统，支持了根据市场价格信息控制饲养成长增重规律，预测最佳上市日期和最好经济效益的优化目标，实现基于个体或小群信息和科学管理决策知识的精细养殖。对性能进行自动化系统测定，利用自动给料站，当耳朵上装有电子感应器的猪走进饲喂站时，接收器立即辨识其耳号，并传送到计算机里记录，测量每次的采食量，并记录该个体每日采食次数、时间、重量等，由计算机完成初步的统计工作。九十年代以来，这些已成熟应用的计算机管理信息系统，正在新的信息技术进步基础上拓宽其高新技术应用研究领域，如：

机器视觉技术用于动物行为监视、识别；挤奶机器人的应用；基于体形图象分析评价优良品种选育与动物体重评估；动物饲养管理多媒体知识咨询系统等都已逐步实用化。BigDutchman 公司研制的控制系统，实现了包括温度、湿度、氨气浓度的环境参数采集，饮水量以及采食量数据的自动采集，畜禽舍温度、通风、喂料的自动控制。

我国畜禽舍环境自动控制技术，“八五”期间，中国农业大学报道以 8098 单片机为核心，开发了以畜禽的日龄为基准的温度、湿度和光照度等多因素环境控制系统的硬件与软件技术，初步实现了畜禽舍环境的智能化控制。“九五”期间，又进一步采用模糊控制理论为核心的人工智能技术，研制成具有模糊控制功能的密闭鸡舍环境控制设备。可对全封闭结构的鸡舍，通过调控鸡舍内部的环境因子，来满足鸡群对生长环境的需求（李保明,2002）。

“九五”期间，江苏省农科院与江苏扬州银河科技公司报道研制鸡舍移动视控系统，（胡肄农等,2002）应用移动摄像技术和远程传输技术，以鸡舍为单元进行全方位、多要素的环境和生产状况监测，实现监视、监听、环境监测，预编程序自动控制通风、光照、清粪，监视图像和测控信息使用调制解调器远程传输，管理者通过监控计算机实时监控生产现场。

总体趋势是运用自动控制技术、传感器技术、网络技术等综合集成，实现畜禽养殖环境的自动化控制、监测、监视、全过程的数字化记录和计算机智能优化决策。

第二章 系统设计的技术理论基础

2.1 面向对象的分析与设计技术

2.1.1 结构化方法

结构化方法是面向系统结构、基于数据流和功能的建模方法。它把现实世界描绘为数据在信息系统中的流动,以及数据流动过程中数据向信息的转换。从总体上看,它是一种强烈依赖于数据流图的自顶向下的建模方法。同时,结构化方法还侧重于系统所实现的功能结构,它应用“问题分割”的思想,自顶向下将系统分解为具有一定层次、执行一定功能的一些逻辑“功能模块”,这些“功能模块”按一定的层次结构有机“结合”起来,形成系统的总体功能层次结构。

结构化方法是以功能和信息数据为中心的解决问题策略,一般采用自顶向下、逐层分解的方法对系统进行分析,由此建立系统的一个自顶向下的分析模型。具有代表性的结构化方法有结构化分析(SA)、结构化设计(SD)、结构化分析与设计技术(SADT)等。

结构化方法因其基于数据流和功能分解的特点,具有严格结构化、可复用、可测量等优点,结构化方法还一定程度上降低了系统开发的费用。结构化方法令系统分析与设计者可最大限度地选择设计系统的结构。特别地,结构化方法有利于从整体、从宏观的角度对所研究的问题进行把握,得出解决问题的关键所在。

但是,结构化方法也有其局限性:功能与数据相分离的思想对客观世界的表达与现实世界很不一样,在对现实世界的认识同程序实现之间始终存在着一道难以逾越的鸿沟(周之英,1999);当系统变化或系统复杂性增加时,自顶向下的功能分解原则极大地限制了系统的可重用性、可维护性和可扩展性;同时“模块”的划分及模块之间的耦合也降低了系统对变化的适应性,特别是对以控制关系为重要特性的系统来说,这种模块耦合作用限制了系统对控制因素变化的适应性(仲辉,2003)。

2.1.2 面向对象的方法

面向对象技术最初是从面向对象的程序设计开始的,它的出现以20世纪60年代Simula语言为标志。面向对象语言Smalltalk出现,进一步发展和完善了面向对象的程序设计语言,从此面向对象也和开发方法开始结合,出现了面向对象的开发方法。20世纪80年代中后期,面向对象系统分析与设计逐渐成熟,被计算机界理解和接受(柴跃廷,1999)。

2.1.2.1 面向对象方法的原理及特征

面向对象的方法是以对象为中心的解决问题策略(Ronald J Norman,1996)。面向对象方法学基本思想是:对问题空间进行自然分割,以更接近人类思维的方式建立问题域模型,

以便对客观实体进行结构模拟和行为模拟，从而使设计出的软件尽可能直接地描述现实世界，构造出模块化的、可重用的、维护性好的软件，同时限制软件的复杂性和降低开发维护费用。

(1) 抽象：抽象就是忽略一个主题中与当前目标无关的方面，注意与当前目标有关的方面。比如，我们要设计一个学生成绩管理系统，考察学生这个对象时，我们只关心他的班级、学号、成绩等，而不用去关心他的身高、体重这些信息。抽象包括两个方面，一是过程抽象，二是数据抽象。过程抽象是指任何一个明确定义功能的操作都可被使用者看作单个实体。数据抽象定义了数据类型和施加于该类型对象上的操作。

(2) 继承：继承是一种联结类的层次模型，并且允许和鼓励类重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，新类继承原始类的特性。派生类可以从它的基类继承方法和实例变量，并且新类可以修改或增加新的方法使之更适合特殊需要。继承性很好地解决了软件的可重用性问题（冯玉琳等,1998）。

(3) 封装：是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象（Cay Horstmann,2004）。一旦定义了一个对象的特性，则有必要决定这些特性的可见性，即哪些特性对外部世界是可见的，哪些特性用于表示内部状态。封装保证了模块具有较好的独立性，使得程序维护修改较为容易。

(4) 多态性：指允许不同类的对象对同一消息作出响应。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好地解决了函数同名问题。面向对象程序设计的四大特性，使得软件开发时间缩短，效率高，可靠性好，所开发的程序更强壮，应用程序更易于维护、更新和升级。

2.1.2.2 几种典型的面向对象方法

(1) OMT/RumbaughOMT(Object Modeling Technique)方法最早是由 Loomis, Shan 和 Rumbaugh 在 1987 年提出的。这个方法是在实体关系模型上扩展了类、继承和行为而得到的。OMT 覆盖了分析、设计和实现三个阶段，包含分析、系统设计、对象设计和实现四个步骤，它定义了对象模型、动态模型和功能模型三种模型，这些模型贯穿于每个步骤，并在每个步骤中被不断地细化和扩充。

(2) OOD/BoochOOD(Object Oriented Design)方法是 GradyBooch 从 1983 年开始研究，1991 年后走向成熟的一种方法。Booch 方法在面向对象的设计中主要强调多次重复和开发者的创造性。方法本身是一组启发性的过程式建议。OOD 的一般过程如下：在一定抽象层次上标识类与对象；标识类与对象的语义；标识类与对象之间的关系(如继承、实例化、使用等)；实现类与对象。

(3) RDD/WirfsBrockRDD(Responsibility Driven Design)方法是 Wirfs Brock 在 1990 年提出的。这是一个按照类、责任以及合作关系对应用进行建模的方法。首先定义系统的类与对象，然后确定系统的责任并划分给类，最后确定对象类之间的合作来完成类的责任。这些设计将进一步按照类层次、子系统和协议来完善。RDD 分探索阶段和精化阶段，按照类层次图、合作图、类规范、子系统规范、合同规范等设计规范来完成实现。

(4) OOAD/CoadYourdonOOAD(Object Oriented Analysis and Design)方法是由 PeterCoad 和 EdwardYourdon 在 1991 年提出的。这是一种逐步进阶的面向对象建模方法。在 OOA 中, 分析模型用来描述系统的功能, 它使用了基本的结构化原则, 并把它们同面向对象的观点结合起来。OOA 完成系统分析, 包括以下五个步骤: 确定类与对象, 标识结构, 定义主题, 定义属性和定义服务。OOA 本质上是一种面向对象的方法, 适用于小型系统的开发。OOD 负责系统设计, 包括以下四个步骤: 设计问题域(细化分析结果); 设计人机交互部分(设计用户界面); 设计任务管理部分(确定系统资源的分配); 设计数据管理部分(确定持久对象的存储)。

(5) OOSE/JacobsonOOSE(Object Oriented Software Engineering)是 IvarJacobson 在 1992 年提出的一种使用事例驱动的面向对象开发方法。OOSE 开发过程中有以下五种模型, 这些模型是自然过渡和紧密耦合的: 需求模型从用户的观点上完整地刻画系统的功能需求。分析模型是在需求模型的基础上建立的。主要目的是要建立在系统生命期中可维护、有逻辑性、健壮的结构。设计模型进一步精化分析模型并考虑了当前的实现环境。实现模型主要包括实现块的代码。测试模型包括不同程度的保证。

(6) VMT/IBMVMT(Visual Modeling Technique)方法是 IBM 公司于 1996 年公布的。VMT 方法结合了 OMT、OOSE、RDD 等方法的优点, 并且结合了可视化编程和原型技术。VMT 方法选择 OMT 方法作为整个方法的框架, 并且在表示上也采用了 OMT 方法的表示。VMT 方法用 RDD 方法中的 CRC(Class Responsibility Collaboration)卡片来定义各个对象的责任(操作)以及对象间的合作(关系)。此外, VMT 方法引入了 OOSE 方法中的使用事例概念, 用以描述用户与系统之间的相互作用, 确定系统为用户提供的服务, 从而得到准确的需求模型(姜鸿飞等,1998)。

2.1.3 面向对象方法与结构化方法的比较

(1) OMT 方法覆盖了应用开发的全过程, 是一种比较成熟的方法, 用几种不同的观念来适应不同的建模场合, 它在许多重要观念上受到关系数据库设计的影响, 适合于数据密集型的信息系统的开发, 是一种比较完善和有效的分析与设计方法(李光志等,1996)。

(2) Booch 方法并不是一个开发过程, 只是在开发面向对象系统时应遵循的一些技术和原则。Booch 方法是从外部开始, 逐步求精每个类, 直到系统被实现。因此, 它是一种分治法, 支持循环开发, 它的缺点在于不能有效地找出每个对象和类的操作。

(3) RDD 是一种用非形式的技术和指导原则开发合适的设计方案的设计技术。它用交互填写 CRC 卡片的方法完成设计, 对大型系统设计不太适用。RDD 采用传统的方法确定对象类, 有一定的局限性。另外, 均匀地把行为分配给类也十分困难。

(4) 在 OOAD 方法中, OOA 把系统横向划分为五个层次, OOD 把系统纵向划分为四个部分, 从而形成一个清晰的系统模型。OOAD 适用于小型系统的开发。

(5) OOSE 能够较好地描述系统的需求, 是一种实用的面向对象的系统开发方法, 适合于商务处理方面的应用开发。

(6) VMT 基于现有面向对象方法中的成熟技术,采用这些方法中最好的思想、特色、观点以及技术,并把它们融合成一个完整的开发过程。因此 VMT 是一种扬长避短的方法,它提供了一种实用的能够处理复杂问题的建模方法和技术。

选用面向对象的方法设计和开发,能有效的建立起问题域模型,真实反映客观实体,构造出模块化的、可重用的、维护性好的软件,降低开发维护费用。

2.2 组件技术

2.2.1 J2EE 技术框架介绍

2.2.1.1 J2EE 的概念

J2EE (Java 2 Platform Enterprise Edition) 是 Java 语言的发明者 Sun 公司提出的一套技术标准(规范),该标准详细地定义了采用 Java 语言实现大型多层分布式应用系统各个组成部分之间的接口规范。

J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版, J2EE 不仅巩固了标准版中的许多优点,例如“编写一次、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等等,同时还提供了对 EJB(Enterprise JavaBeans)、Java Servlet API、JSP (Java Server Pages) 以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE 体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性,同时提供对现有应用程序集成强有力支持,完全支持 Enterprise JavaBeans,有良好的向导支持打包和部署应用,添加目录支持,增强了安全机制,提高了性能。

2.2.1.2 J2EE 的核心 API 组件

J2EE 平台由一整套服务 (Services)、应用程序接口 (APIs) 和协议构成,它对开发基于 Web 的多层应用提供了功能支持,下面对 J2EE 中的一些技术规范进行简单的描述(限于篇幅,这里只能进行简单的描述):

(1) JDBC (Java Database Connectivity)

JDBC API 为访问不同的数据库提供了一种统一的途径,象 ODBC 一样, JDBC 对开发者屏蔽了一些细节问题,另外, JDBC 对数据库的访问也具有平台无关性。

(2) JNDI (Java Name and Directory Interface)

JNDI API 被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源如 DNS 和 LDAP,本地文件系统,或应用服务器中的对象。

(3) EJB (Enterprise JavaBean)

J2EE 技术之所以赢得广泛重视的原因之一就是 EJB (陈华军,2002)。它们提供了一个框架来开发和实施分布式商务逻辑,由此很显著地简化了具有可伸缩性和高度复杂的企业级应用的开发。EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用。容器负

责提供公用的服务，例如目录服务、事务管理、安全性、资源缓冲池以及容错性。但这里值得注意的是，EJB 并不是实现 J2EE 的唯一途径。正是由于 J2EE 的开放性，使得有的厂商能够以一种和 EJB 平行的方式来达到同样的目的。

(4) RMI (Remote Method Invoke)

正如其名字所表示的那样，RMI 协议调用远程对象上的方法。它使用了序列化方式在客户端和服务端传递数据。RMI 是一种被 EJB 使用的更底层的协议。

(5) JSP (Java Server Pages)

JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端所请求以后对这些 Java 代码进行处理，然后将生成的 HTML 页面返回给客户端的浏览器。

(6) Java Servlet

Servlet 是一种小型的 Java 程序，它扩展了 Web 服务器的功能。作为一种服务器端的应用，当被请求时开始执行，这和 CGI Perl 脚本很相似。Servlet 提供的功能大多与 JSP 类似，不过实现的方式不同。JSP 通常是大多数 HTML 代码中嵌入少量的 Java 代码，而 servlets 全部由 Java 写成并且生成 HTML。

(7) JMS (Java Message Service)

JMS 是用于和面向消息的中间件相互通信的应用程序接口(API)。它既支持点对点的域，有支持发布/订阅(publish/subscribe)类型的域，并且提供对下列类型的支持：经认可的消息传递，事务型消息的传递，一致性消息和具有持久性的订阅者支持。JMS 还提供了另一种方式来对您的应用与旧的后台系统相集成。

(8) JTA (Java Transaction Architecture)

JTA 定义了一种标准的 API，应用系统由此可以访问各种事务监控。

(9) JTS (Java Transaction Service)

JTS 是 CORBA OTS 事务监控的基本的实现。JTS 规定了事务管理器的实现方式。该事务管理器是在高层支持 Java Transaction API (JTA)规范，并且在较底层实现 OMG OTS specification 的 Java 映像。JTS 事务管理器为应用服务器、资源管理器、独立的应用以及通信资源管理器提供了事务服务。

(10) JavaMail

JavaMail 是用于存取邮件服务器的 API，它提供了一套邮件服务器的抽象类。不仅支持 SMTP 服务器，也支持 IMAP 服务器。

2.2.1.3 J2EE 应用的四层结构组件

J2EE 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据他们所在的层分布在不同的机器上。事实上，Sun 设计 J2EE 的初衷正是为了解决两层模式的弊端。在传统模式中，客户端担当了过多的角色而显得臃肿，在这种模式中，第一次部署的时候比较容易，但难于升级或改进，可伸展性也不理想，而且经常基于某种专有的协议——通常是某种数据库协议。它使得重用业务逻辑和界面逻辑非常困难。现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。一个多层化应用能够为不同的每种服务提供一个独立的层，J2EE 典型的四层结构：运行在客户端机器上的客户层

组件、运行在 J2EE 服务器上的 Web 层组件、运行在 J2EE 服务器上的业务逻辑层组件、运行在 EIS 服务器上的企业信息系统(Enterprise information system)层软件。J2EE 应用的多层结构如图 2.1 所示。

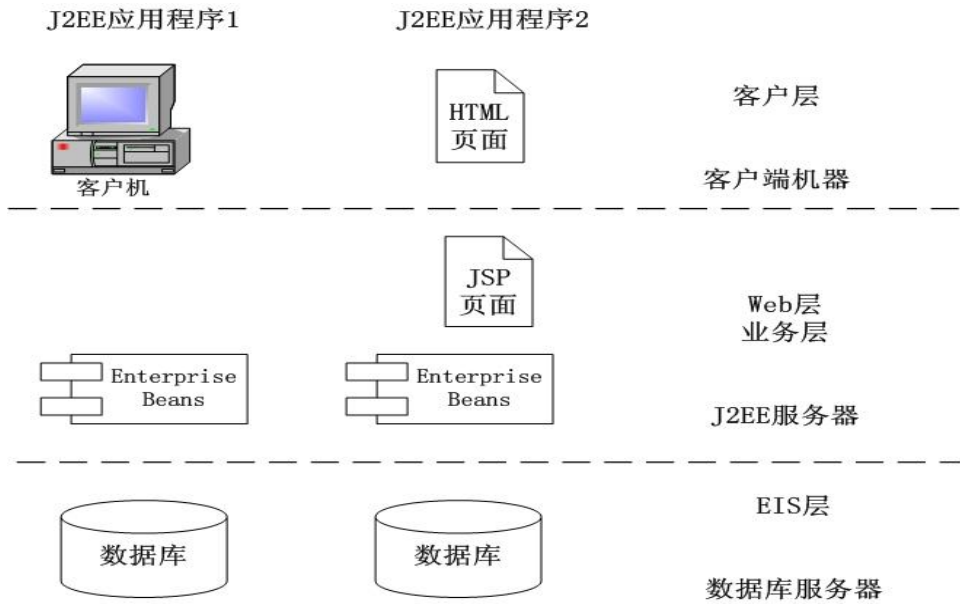


图 2.1 J2EE 应用的多层结构

Fig.2.1 The multilayer construction of J2EE application

J2EE 应用程序是由组件构成的。J2EE 组件是具有独立功能的软件单元，它们通过相关的类和文件组装成 J2EE 应用程序，并与其他组件交互。J2EE 说明书中定义了以下的 J2EE 组件：

应用客户端程序和 applets 是客户层组件。

Java Servlet 和 JavaServer Pages(JSP)是 web 层组件。

Enterprise JavaBeans(EJB)是业务层组件。

(1) 客户层组件

J2EE 应用程序可以是基于 web 方式的，也可以是基于传统方式的。

(2) web 层组件

J2EE web 层组件可以是 JSP 页面或 Servlets。按照 J2EE 规范，静态的 HTML 页面和 Applets 不算是 web 层组件。

如图 2.2 所示的客户层那样，web 层可能包含某些 JavaBean 对象来处理用户输入，并把输入发送给运行在业务层上的 Enterprise Bean 来进行处理。

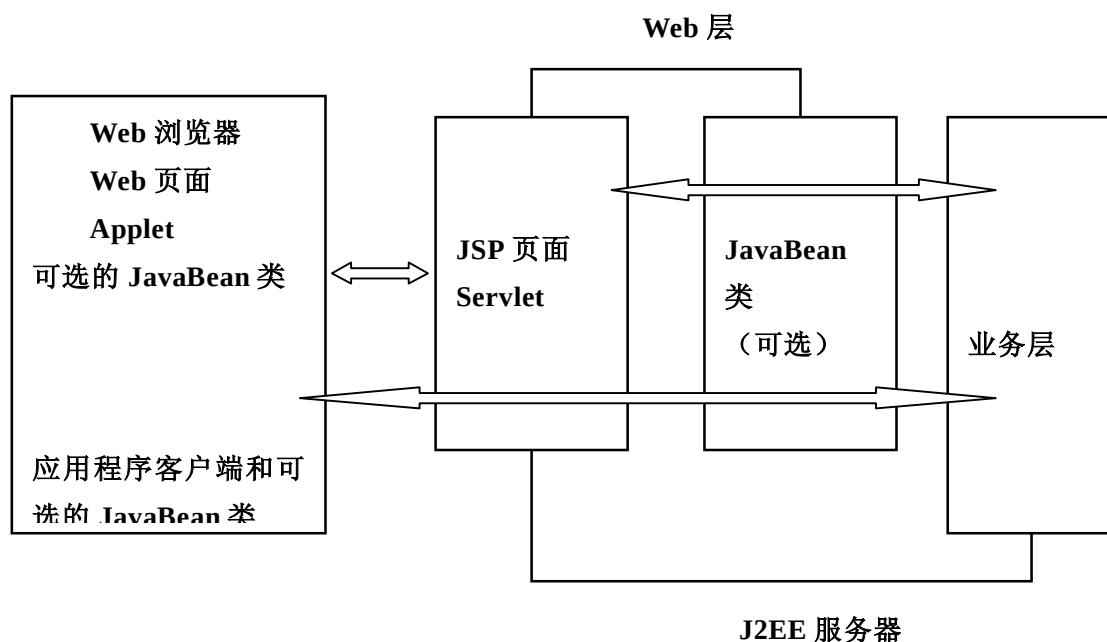


图 2.2 web 层组件

Fig.2.2 The component of Web tier

(3) 业务层组件

业务层代码的逻辑用来满足银行，零售，金融等特殊商务领域的需要，由运行在业务层上的 enterprise bean 进行处理。图 2.3 表明了一个 enterprise bean 是如何从客户端程序接收数据，进行处理(如果必要的话)，并发送到 EIS 层储存的，这个过程也可以逆向进行。

有三种企业级的 bean：会话(session) beans，实体(entity) beans，和消息驱动(message-driven) beans。会话 bean 表示与客户端程序的临时交互。当客户端程序执行完后，会话 bean 和相关数据就会消失。相反，实体 bean 表示数据库的表中一行永久的记录。当客户端程序中止或服务器关闭时，就会有潜在的服务保证实体 bean 的数据得以保存。消息驱动 bean 结合了会话 bean 和 JMS 的消息监听器的特性，允许一个业务层组件异步接收 JMS 消息。

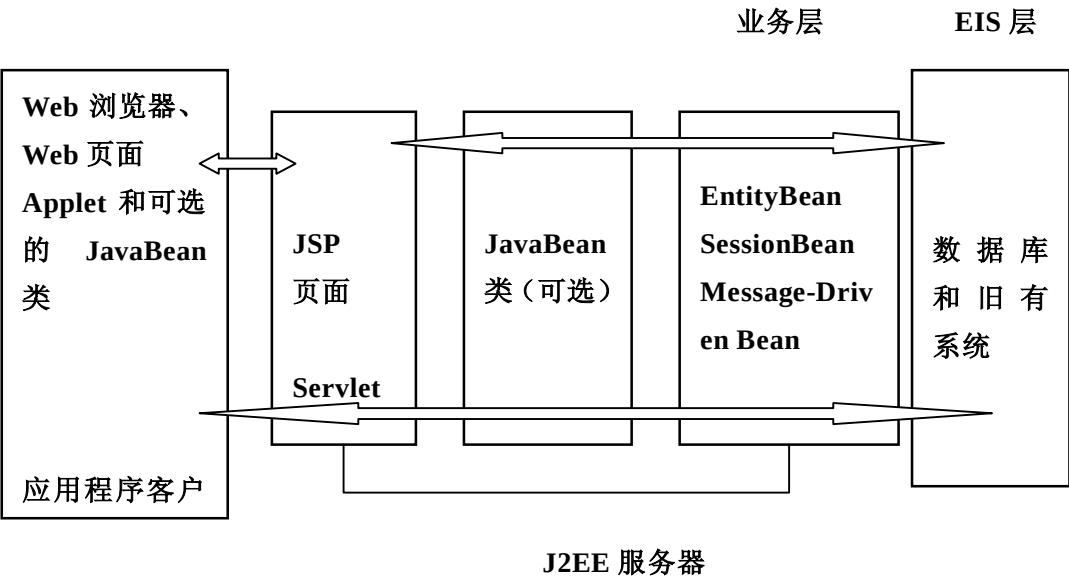


图 2.3 业务层组件

Fig.2.3 The component of operation tier

(4) 企业信息系统层

企业信息系统层处理企业信息系统软件，包括企业基础建设系统例如企业资源计划 (ERP)，大型机事务处理，数据库系统和其它的遗留信息系统。例如，J2EE 应用组件可能为了数据库连接需要访问企业信息系统。

2.2.2 .NET 技术框架介绍

Microsoft.NET 战略基于一组开放的互联网协议，推出了一系列的技术、产品和服务，吹响了一次互联网技术变革的号角。图 2.4 所示的.NET 开发框架几个主要组成部分：首先是整个开发框架的基础，即通用语言运行时以及它所提供的一组基础类库；在开发技术方面，.NET 提供了全新的数据库访问技术 ADO.NET，以及网络应用开发技术 ASP.NET 和 Windows 编程技术 Win Forms；在开发语言方面，.NET 提供了 Visual Basic、Visual C++、C#和 Javascript 等多种语言支持；而 Visual Studio.NET 则是全面支持.NET 的开发工具。

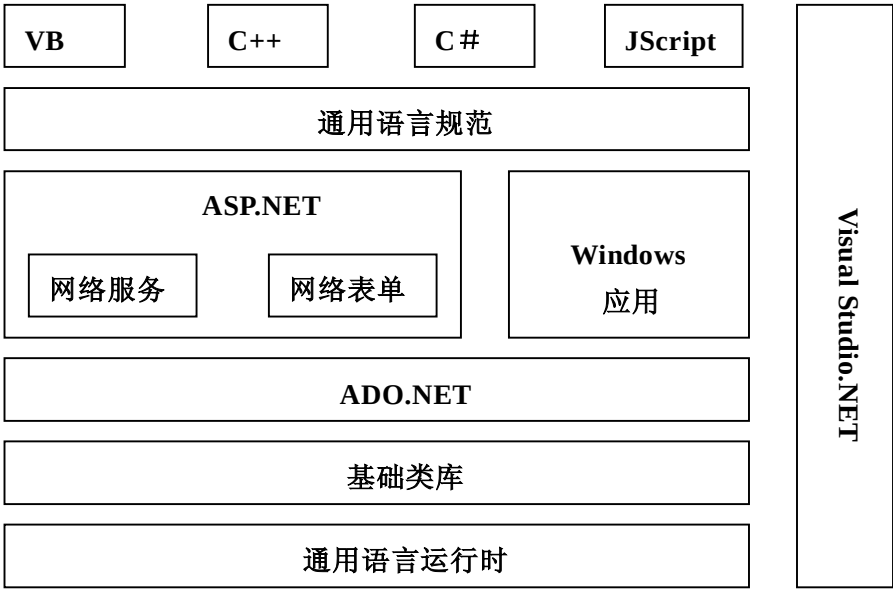


图 2.4 .NET 开发框架

Fig.2.4 The Development frame of .NET

通用语言运行时环境，即 Common Language Runtime。Microsoft.NET 给开发人员带来了一种全新的开发框架，而通用语言运行时环境则处于这个框架的最低层，是这个框架的基础。开发人员对于所谓的 C 运行时、Visual Basic 运行时、Java 虚拟机这些概念已经非常熟悉了，而通用语言运行时则为多种语言提供了一种统一的运行环境。另外它还提供了更多的功能和特性，比如统一和简化的编程模型，用户不必迷惑于 Win32 API 和 COM；避免了 DLL 的版本和更新问题（常称为 DLL 地狱），从而大大简化了应用程序的发布和升级；多种语言之间的交互，例如我们甚至可以在 Visual Basic 中使用 C++ 编写的类；自动的内存和资源管理等等。Microsoft.NET 正是基于通用语言运行时，实现了这些开发人员梦寐以求的功能。

基于通用语言运行时开发的代码称为受控代码，它的运行步骤大体如下：首先使用一种通用语言运行时支持的编程语言编写源代码，然后使用针对通用语言运行时的编译器生成独立于机器的 Microsoft 中间语言（Microsoft Intermediate Language, MIL），同时产生运行所需的元数据，在代码运行时再使用即时编译器（Just In Time Compiler）生成相应的机器代码来执行。

当然对于开发者而言，他们除了关心通用语言运行时提供那么多新特性外，它究竟给开发者提供了什么样的编程接口，这就是基础类库（Base Class Library）。这组基础类库包括了从输入输出到数据访问等各方面，提供了一个统一的面向对象的、层次化的、可扩展的编程接口。它使用一种点号分隔的方法，使得查找和使用类库非常容易。

2.2.3 两大技术体系比较

在表现形式上，J2EE 是一组规范，而.NET 更象是一组产品。但它们的目的是为了企业应用提供分布式的，高可靠性的解决方案。它们在架构上有着很多的相似之处，表 2.1 是一个简单对照：

表 2.1 J2EE 与.NET 两大技术架构比较

Tab2.1 The technology constructions comparison of J2EE &.NET

	J2EE	.NET
通信协议	Remote Method Invocation over Internet InterOrb Protocol (RMI/IIOP), XML	
编程语言	Java	C#, VB.NET, COBOL
运行时环境	Java Virtual Machine (JVM)	Common Language Runtime (CLR)
胖客户端	Java Swing	Windows Forms
目录服务	Java Naming and Directory Interface (JNDI)	Active Directory Services Interface (ADSI)
数据访问	Java Database Connection (JDBC) , Java Connectors	ADO.NET
异步消息处理	Java Message Service (JMS)	Microsoft Message Queue
表示层技术	Servlets, Java Server Page(JSP)	ASP.NET
中间层组件模型	EJB, JavaBean	COM+, COM
安全访问	JAAS	COM+ Security
事物处理	Java Transaction Server (JTS)	Microsoft Distributed Transaction Coordinator (MS-DTC)
开发工具	WebGain Visual Cafe Borland JBuilder IBM VisualAge 等 (第三方提供，规范本身没有定义)	Visual Studio.NET

整体评价（见表 2.2）：
内部众多系统的整合、系统的延展性、安全性是 J2EE 的优势，也是.NET 的不足处。在效率方面，J2EE 阵营主张通过硬件的效能增加来弥补软件的不足.开放标准，功能强大，易于移植这些都是 J2EE 的优点。

表 2.2 两大技术整体评价

Tab2.2 The whole appraisal of J2EE & .NET

	J2EE	.NET
易用性	**	***
扩展能力	***	**
多平台支持	****	*
多语言支持	*	****
可靠性	***	***
性能	***	***
可管理性	***	***
重用性	****	**
负载平衡	***	***
开放标准	*****	*

2.3 web 开发技术

2.3.1 C/S 模式 、B/S 模式和混合模式

2.3.1.1 C/S 模式

C/S(Client/Server)结构即客户机/服务器系统，它是由一个或多个客户和一个或多个服务器与下层的操作系统和网络系统进行信息交互。结构分为两层结构和三层结构两种，其中两层结构即胖客户端是最典型、也是最普遍的一种形式。这种形式的 C/S 结构分为两层：第一层是在客户机系统上结合了用户界面与业务逻辑(在客户端程序里)；第二层是通过网络结合了数据库服务器。在 C/S 两层结构中，客户端保持着应用程序，直接访问数据库；服务器端存放着所有数据，每个客户与数据库保持一个信任连接。客户端通过应用程序向数据服务器发出 SQL 请求，数据库服务器据此请求对数据库进行操作，并向客户端返回应答结果。

C/S 模式体系结构如图 2.5 所示：

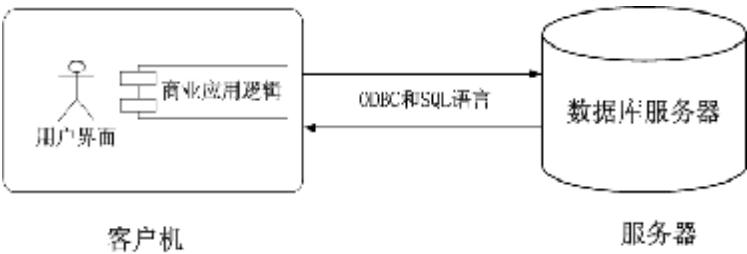


图 2.5 两层 C/S 结构

Fig.2.5 The C/S construction of two tiers'

两层结构应用软件的开发工作主要集中在客户方，客户方软件不但要完成用户交互和数据显示的工作，而且还要完成对应用逻辑的处理工作，既用户界面与应用逻辑位于同一个平台上。这样就带来了两个突出的问题：系统的可伸缩性较差和安装维护困难，使得两层结构应用软件在 Internet/Intranet 环境下的使用受到较大限制。为了解决两层结构应用软件中存在的问题，人们又提出了多层结构应用软件。

在三层结构应用软件中，整个系统由三个部分组成：客户机、应用服务器和数据库服务器。客户机(Client)只需要安装应用程序，它通常实现用户界面，提供可视化的用户接口，负责用户和应用程序间的对话，用来表示信息和收集数据，负责处理与用户的交互和与应用服务器的交互。

应用服务器(Application Server)是应用逻辑处理的核心，是连接客户和数据库服务器的桥梁。应用服务器和客户间的数据交往应当尽量简捷，客户将请求信息发送给应用服务器，应用服务器负责处理应用逻辑，具体地说就是接受客户方应用程序的请求，然后根据应用逻辑将这个请求转化为数据库请求后与数据库服务器交互，并将与数据库服务器交互的结果传给客户机方的应用程序。数据库服务器(Database Server)以传统的基于 SQL 的 DBMS 实现，负责管理数据的定义、维护、访问和更新，并响应应用服务器的数据请求。软件根据应用服务器发送的请求进行数据库操作，并将操作的结果传给应用服务器。

三层结构应用软件的模型如图 2.6 所示：



图 2.6 三层 C/S 结构

Fig.2.6 The C/S construction of three tiers

由于在三层 C/S 结构中，应用服务器实现应用服务，数据库服务器相连，而只和应用服务器相连，通过应用服务器和数据库服务器相连。因此，与两层 C/S 结构相比三层 C/S 具有如下明显的优势。

2.3.1.2 B/S 模式体系结构

B/S (Browser/Server)结构指浏览器/服务器系统，它实际上是以浏览器作为标准前端进行信息交互。它是一种三层结构的系统：第一层客户机是一个通用的浏览器；用户在浏览器的网页上输入信息提出处理请求提交给后台，这个后台就是第二层的，Web 服务器。Web 服务器将启动相应的进程来响应这一请求，并将处理的结果返回给浏览器。第三层数据库服务器同 C/S 模式类似，负责管理数据库。

B/S 模式中网络的典型拓扑结构如图 2.7 所示：

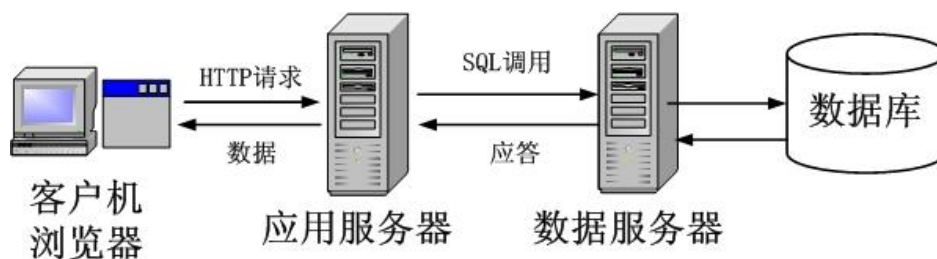


图 2.7 B/S 模式中网络的典型拓扑结构

Fig.2.7 The typical B/S construction of network topology

这种系统在服务器端集中了所有应用逻辑，在客户机上通过直观、易于使用的浏览器来从 Web 服务器上获取信息。Web 服务器通过 HTTP 建立了内部页面和相关后端数据库的超文本连接，所以最终可以用浏览器查询所有网络服务器上信息。其中浏览器构成了系统的表示层，Web 服务器及应用服务器构成了系统的应用层，而数据库服务器构成了系统的数据层，因此 B/S 模式的信息系统由表示层、应用层和数据层组成。

在 B/S 模式中，客户端对服务端运行的程序并不是直接调用而是通过 Web 服务器进行协调，由 Web 服务器向服务端程序转交客户端的调用申请，当服务器程序允许调用时，将复制一份客户端副本经 Web 服务器反馈回客户端并在服务端保留响应的申请记录。数据库系统只在服务器监控各个副本的运行情况并随时响应有效的客户申请和操作。客户端只对服务端程序所复制的客户端副本进行本地解释和操作。任何操作响应都是在客户端由副本完成的，当副本向服务端程序申请数据更新或服务端程序因数据更新而对整个网络中的客户端进行更新广播时，系统只将更新的部分经由 Web 服务器与客户端进行交互，而未更新部分则保持旧有的运行状态不变而不被交换，因此网络传输的只是被更新的信息，从而使通信总量大为降低，当用户退出数据库系统时客户端副本自动丢弃，服务端程序删除副本的相关记录并终止监控。在对数据的操作过程中，客户端与服务器始终是由 Web 服务器隔离的，这样不会因某个客户端的系统崩溃或错误操作而影响整个应用系统的正常运行，从而增加了对数据操作的技术安全性。

B/S 模式的开放性还允许用户在网络中的不同位置 and 不同客户端上随时申请对信息的操作，而无须担心因在网络中所处物理位置的变化或所使用的终端不同而被客户端程序拒绝访问。用户也无须了解数据库系统在网络中的运行环境和存储结构，用户只须拥有有效权限就可以访问服务器终端程序，而这种授权机制是可以无缝连接的，因此用户成功登录后其在网络中的逻辑位置将是透明的。

C/S 模式与 B/S 模式有着各自结构上的优缺点：C/S 模式可以实现数据的快速传送和安全存储，但在远程数据管理方面存在不足；B/S 模式的系统易于集成和扩展，可以提供远程的数据服务和管理，但在数据的交互性、动态服务和图表显示方面尚有欠缺。

C/S 结构和 B/S 结构的区别如下：

硬件环境不同：C/S 一般建立在专用的网络上，小范围里的网络环境，局域网之间再通过专门服务器提供连接和数据交换服务；B/S 建立在广域网之上的，不必是专门的网络

硬件环境，例如电话上网，租用设备。信息自己管理，有比 C/S 更强的适应范围，一般只要有操作系统和浏览器就可以。

安全要求不同：C/S 一般面向相对固定的用户群，对信息安全的控制能力很强。一般高度机密的信息系统采用 C/S 结构。可以通过 B/S 结构发布部分可以公开信息，B/S 建立在广域网之上，对安全的控制能力相对较弱，面向是不可知的用户群。

程序架构不同：C/S 程序可以更加注重流程，可以对权限多层次校验，对系统运行速度考虑较少。B/S 对安全以及访问速度的考虑较多，建立在需要更加优化的基础之上。

软件重用不同：C/S 程序可以不可避免的整体性考虑，构件的重用性不如在 B/S 要求下的构件的重用性好。

系统维护不同：C/S 程序由于整体性考虑，处理出现问题以及系统升级可能是再做一个全新的系统。B/S 构件组成方便个别的更换，实现系统的无缝升级，系统维护开销小。

处理问题不同：C/S 程序面向固定用户，并且在相同区域，要求安全性好并且与服务器操作系统相关。B/S 建立在广域网上，面向不同的用户群与操作系统关系小。

用户接口不同：C/S 多是建立在 Windows 平台上，表现方法有限，对程序员普遍要求较高。B/S 建立在浏览器上，有更加丰富和生动的表现方式，减低开发成本。

现对 B/S 模式、C/S 模式以及二者结合的模式在实际信息系统中的几个评价指标分析如表 2.3:

表 2.3 B/S 模式、C/S 模式以及二者相结合的模式指标评价

Tab2.3 B/S pattern, C/ S pattern as well as the two unify pattern indices evaluation

评价指标	B/S 模式	C/S 模式	二者结合
交互性	较弱	强	较高
安全性	低	高	较高
网络流量	大	小	较小
通信速度	慢	快	较快
开发难度	较大	小	大

相对于单独采用 C/S 或 B/S 模式，混合模式方案的优点在于：

- (1) 保证敏感数据的安全性，特别是对数据库修改和新增记录加强控制；
- (2) 有效利用企业内部计算机的资源，降低部分客户端机器的配置；
- (3) 既保证了复杂功能的交互性，又保证了一般功能的易用与统一；
- (4) 系统维护简便，布局合理，网络效率高。

综上所述，C/S 模式可以实现分布的处理与集中的数据管理相结合，同时数据安全性强，通信速度快；B/S 模式则有利于实现系统集成与扩展。因此，企业在构建自己的信息系统时，在安全性要求高、交互性强、处理数据量大、数据查询灵活的地点固定的小范围内使用 C/S 模式，在安全性和交互性要求不高，地点灵活的广域范围内使用 B/S 模式，将二种模式相结合，充分发挥各自的长处，这样可以开发出安全可靠、灵活方便、效率高的软件系统。

2.3.2 目前流行的 web 开发语言介绍

Web 应用程序就是 Web 服务器的扩展, 随着 Web 的不断发展, 开发 Web 应用程序的编程语言工具也不断推陈出新。当前流行的 Web 开发技术主要有 Microsoft 的 ASP, 自由软件 PHP 和 Sun 的 JSP 等技术。

2.3.2.1 ASP 技术

ASP(Active Server Pages)是由微软创建的 Web 应用开发标准。ASP 服务器已经包含在 IIS 服务器中, ASP 服务器将 Web 请求转入解释器中, 在解释器中将所有 ASP 中的脚本进行分析, 然后执行, 同时可以创建 COM 对象以完成更多的功能。ASP 默认的脚本编写语言为 VBScript, 但是用户可以根据自己的需要进行选择。只要用户安装了相应的脚本引擎, 就可以使用几乎目前所有流行的脚本语言编写 ASP 应用程序。

ASP 的核心技术是对组件和对象技术的充分支持。通过使用 ASP 的组件和对象技术, 用户可以直接在 ASP 页面中使用 ActiveX 控件、Visual Basic 程序就是 Web 服务器的扩展, 随着 Web 的不断发展, 开发 Web 应用程序的编程语言工具也不断推陈出新, C++ 的 COM 对象等, 调用对象方法和属性, 以简单的方式实现强大的功能。

通过使用第三方开发的各种 COM 对象, 可以大大的节省开发人员编写代码的数量和时间。

ASP 的优点是: 安装配置方便, 开发简单易学; 开发工具功能强大。不足之处在于: ASP 使用了组件因而将导致大量的安全问题; 无法实现跨平台, 只能应用于 Windows NT/2000。一般采用微软架构的网站采用 ASP 应用开发平台。

2.3.2.2 PHP 技术

PHP(Hypertext Preprocessor, 即超文本预处理器)是一种 HTML 内嵌式的语言。由于其良好的性能及免费的特点, 是目前互联网应用中很流行的一种应用开发平台。

相对于 ASP, JSP 等其他网络编程语言, PHP 本身是一种脚本语言, 他在很大程度上综合了 Perl, Java 和 C 语言的精华, 在语法结构上继承了 C 语言的风格的同时加入了自己的语法。在数据库方面, PHP 能够支持 Sybase, Oracle, Informix, SQLServer 等多种数据库。PHP 具有良好的跨平台可移植性, 可以在 Windows, Unix 和 Linux 系统下的 Web 服务器上正常运行, 此外, 当用户变换工作平台时, 不需要更改 PHP 代码, 直接就可以在新的平台上使用。

PHP 的优点是: 简单易学、跨平台、有良好数据库交换能力; 与 Apache 及其扩展库紧密结合; 良好的安全性。不足之处在于: 安装配置复杂; 缺少企业级的支持; 作为自由软件, 缺乏正规的商业支持; 无法实现商品化的商业开发。一般基于 Unix 类平台的应用采用 PHP 配合 Apache 使用。

2.3.2.3 JSP 技术

Java Server Pages 是 Sun MicroSystem 公司倡导, 采纳了计算机软硬件、通信、数据库领域多家厂商的意见而共同制定的一种基于 Java 的 Web 动态网页标准语言。JSP 和一般的

Java 语言一样, 具有面向对象性、便利性、跨平台等优点和特性。秉承了 Java 的“一次编写, 到处运行(Write Once, Run Anywhere)”的精神, 既与硬件无关, 也与操作系统和 Web 服务器无关, 是一种与平台无关的技术。JSP 拥有 Java Servlet 的所有稳定性, 另外还可以使用 Servlet 提供的 API。还因为使用了 Java 线程, 比起其它服务器脚本语言, 更适用于大型服务器, 而且开发简单, 更加安全。

JSP 运行于服务器端, 是 Java 阵营中最具有代表性的解决方案。一般来说, 在实际的 JSP 引擎中, JSP 页面在执行时是编译式, 而不是解释式的。JSP 工作原理如图 2.8 所示。

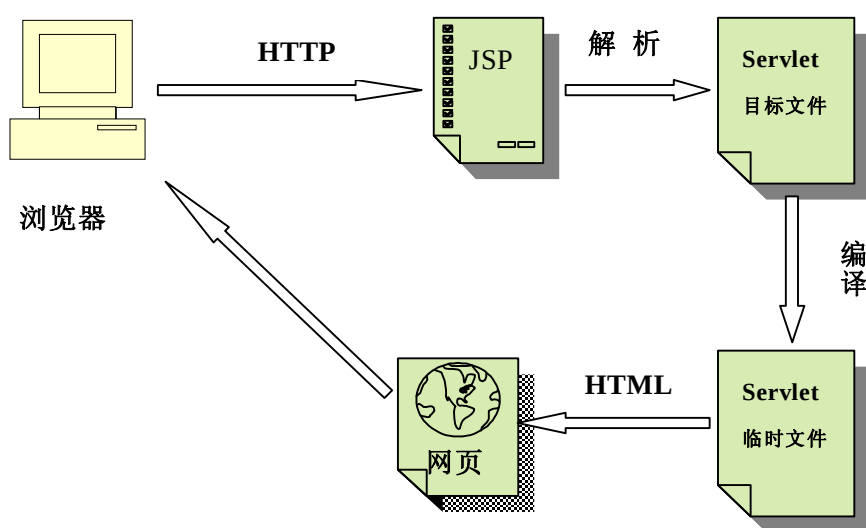


图 2.8 JSP 工作原理

Fig.2.8 The work principle of JSP

JSP 页面首先被解析为 Servlet 的 Java 源文件, 再经过 Java 编译为 Servlet 的 class 文件。将 JSP 编译为 Servlet 是为了让原先的 Servlet 引擎可以直接服务于 JSP, 而 JSP 引擎就仅仅需要将 JSP 转译为 Servlet 就可以了。在 JSP 文件转译为 Servlet 以后, 每次客户机(通常是用户的 Web 浏览器)向服务器请求这一个 JSP 文件的时候, 服务器将检查自上次编译后的 JSP 文件是否有改变, 如果没有改变, 就直接执行 Servlet, 而不用再重新编译, 其效率是相当高的。一般来说, JSP 文件的编译是在第一个用户访问到这个 JSP 页面时发生, 而这第一个用户通常都是开发人员自己, 这样, 正式放在服务器上让用户访问的 JSP 文件一般都是已经有了对应的编译好的 Servlet 了。许多服务器都有设置, 可以使 JSP 文件在第一个用户访问之前就预先编译好, 这样的效率就更高了。

JSP 的优点主要有:

(1) 将内容的产生和显示进行分离

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。使用 JSP 标识或者小脚本来产生页面上的动态内容。产生内容的逻辑被封装在标识和 JavaBeans 群组件中, 并且捆绑在小脚本中, 所有的脚本在服务器端执行。如果核心逻辑被封装在标识和 Beans 中, 那么其它人, 如 Web 管理人员和页面设计者, 能够编辑和使用 JSP 页面, 而不影响内容的产生。在服务器端, JSP 引擎解释 JSP 标识, 产生所请求

的内容（例如，通过存取 JavaBeans 群组件，使用 JDBC 技术存取数据库），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这有助于作者保护自己的代码，而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

（2）强调可重用的群组件

绝大多数 JSP 页面依赖于可重用且跨平台的组件（如 JavaBeans 或 Enterprise JavaBeans）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者用户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种群组织在他们现有的技能和优化结果的开发努力中得到平衡。

（3）采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的程序设计人员。JSP 技术封装了许多功能，这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容产生所需要的。标准的 JSP 标识能够存取和实例化 JavaBeans 组件，设定或者检索群组件属性，下载 Applet，以及执行用其它方法更难于编码和耗时的功能。通过开发定制化标签库，JSP 技术是可以扩展的。第三方开发人员和其它人员可以为常用功能建立自己的标签库，这样 Web 页面开发人员能够使用熟悉的工具和如同标签一样的执行特定功能的构件来工作。JSP 技术很容易整合到多种应用体系结构中，以利用现存的工具和技巧，并且扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 J2EE 的一个成员，JSP 技术能够支持高度复杂的基于 Web 的应用。由于 JSP 页面的内置脚本语言是基于 Java 程序设计语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，JSP 页面就具有 Java 技术的所有好处，包括健壮的存储管理和安全性。作为 Java 平台的一部分，JSP 拥有 Java 程序设计语言“一次编写，各处执行”的特点。随着越来越多的供货商将 JSP 支持加入到他们的产品中，您可以使用自己所选择的服务器和工具，修改工具或服务器并不影响目前的应用。

2.3.2.4 几种技术的比较

ASP 技术只能运行于微软的服务器产品上，而 PHP 可以在 Windows、Unix 和 Linux 的 Web 服务器上运行，还支持 IIS、Apache 等通用的 Web 服务器，用户在更换平台时，需要对代码作一些改变。JSP 几乎可以运行于所有平台，并且从一个平台移植到另一个平台，JSP 程序甚至不用重新编译。在性能比较上有人做过试验，对这三种语言分别作循环性能测试和存取 Oracle 测试，结果 JSP 都是最快的。

另外，PHP 缺乏规模支持 and 多层结构支持，并且提供的数据库接口支持不统一，使它不适合在电子商务中运用。ASP 和 PHP 则没有以上问题，ASP 可以通过 COM/DCOM 获得 ActiveX 规模支持，通过 DCOM 和 Microsoft Transaction Server 获得结构支持，JSP 可以通过 JavaBeans 和 EJB 获得规模支持，通过 EJB/CORBA 以及众多厂商的 Application Server 获得结构支持。表 2.4 列出了 JSP, ASP, PHP 和 CGI 的性能比较。

表 2.4 JSP, ASP, PHP 和 CGI 性能比较

Tab2.4 The performance comparisons of JSP,ASP,PHP & CGI

	JSP	ASP	PHP	CGI
运行速度	快	较快	较快	慢
运行开销	小	较大	较大	较大
平台	均可	Windows	Windows/Unix	均可
扩展性	很好	好	不好	不好
安全性	好	不好	好	好
分布式处理	支持	支持	不支持	不支持
函数支持	广泛	较广泛	广泛	较广泛
数据库支持	多	多	多	多
应用程度	较广泛	较广泛	较广泛	广泛
厂商支持程度	广泛	少	较广泛	广泛
对 XML 的支持	支持	不支持	支持	不支持

由表 2.4 可以看出 JSP 技术在运行速度、开销、平台扩展性、安全性等方面比其它技术存在较大的优势。作为目前一种流行的技术，尤其是在电子商务类的 Web 应用方面，越来越多的使用了 JSP 技术。

2.3.3 Java 语言介绍

自从 SUN 公司 1995 年 5 月发布 Java 以来，在短短几年时间，Java 已风靡全球，成为最热门的程序设计语言，除了它适应形势发展的趋向之外， 必然还有其优秀的特点。Java 语言的特点主要体现在简单(simple)、面向对象(object-oriented)、解释性(interpreted)、分布式 (distributed)、 结构中 立 (architecture-neutral)、 强 壮 (robust), 安 全 (secure), 高 性 能 (high-performance)、 多 线 程 (multithreaded), 可 移 植 (portable)、 动 态 (dynamic) 和 通 用 (general-purpose)等。

1、简单

由于设计者的仔细考虑，使得 Java 语言简单有效，Java 简单性主要从以下三个方面体现：

Java 的风格与 C++相似，设计者考虑到原来 C++应用很广，大多数程序员都熟悉 C++，所以让 Java 程序的外观看起来很象 C++的风格，因此 C++程序员可以很快的掌握 Java 编程技术；

Java 中去掉了 C++中较困难以及容易引发程序错误的部分，如指针(pointer)和内存管理(memory management)、多重继承等。熟悉 C++的人都知道，指针是 C++中较复杂的数据类型，稍不注意会引起程序出错，而且由程序员考虑存储器管理等复杂问题也增加编程困难；

Java 丰富的类库提供的基本功能使编程人员能快速有效地开发应用程序，从而简化开发过程。

2、面向对象

Java 最重要的特点是它是真正面向对象的语言。C++不支持类似 C 那样的面向过程的程序设计技术，它是在 C 的基础上改造成面向对象的，而 Java 从开始设计时就面向对象技术来设计。它的封装性(Encapsulation)，继承性(Inheritance)，多态性(Polymorphism) 等都很好，还实现了 C++无法实现的动态联编(Dynamic binding)。使得 Java 具有静态和动态代码继承和重用(Inheritance and Reusability of code)，扩展(Extensibility)和动态应用(Dynamic application)等优点。

在 Java 中，面向对象编程的基本元素是对象(object)，它把变量(variable)和方法(method)从功能上封装成一个可重用、可动态装载的整体。类(class)是对象的集合，同一类的对象有同样的方法和性质。

3、分布式

Java 语言是很适合分布式应用的。通过类库(class library)提供的方法，Java 可方便地适应 TCP/IP 协议(Transmission Control Protocol/Internet Protocol，这是 Internet 的通用网络协议)。如 HTTP(HyperText Transfer Protocol，用于 www)和 FTP(File Transfer Protocol，用于 Internet 传送文件)等，Java 应用程序可通过 URL(Universal Resource Locator)存取网络上的对象，其访问方式和访问本地文件系统几乎完全相同。

4、解释性

Java 是解释执行的，但它与 Basic 语言的解释不同。Basic 没有编译阶段。语句输入后，一边检查错误一边运行，效率很低。Java 则是经过编译后产生字节码(Bytecode)，然后由各种支持 Java 解释器(interpreter)的机器来解释执行字节码。由于 Java 的字节码比较接近汇编指令，所以可以很快解释为实际机器指令，这种做法比 Basic 的解释效率高得多。Java 编译后直接解释运行，比 C++少了一个连接阶段，缩短了开发周期。

5、健壮

Java 的强壮性保证代码有好的表现，即稳定可靠地运行。因为 Java 的发明者发现了 C++的不稳定之处：C++用指针直接访问存储器位置，而且没有自动内存管理，即使程序员写出语法和语义都正确的程序，也会由于访问内存出错等原因而造成系统崩溃。因此，Java 用消除指针和提供自动内存管理来保证强壮的运行环境。消除指针后程序员就不能用指针算法在内存中任意移动，防止对操作系统的内存空间进行误写。另外，Java 增加了数组边界检查功能，这样一来，程序就不能访问数组没有分配的寻址空间。

Java 的自动内存管理功能一方面减轻程序员注意内存情况的劳动，另一方面提高内存使用效率，同时防止人为干预内存而造成破坏。

此外，Java 在编译和运行程序时都对程序进行检查，防止错误隐藏。Java 的编译检查比 C++严格，它有异常(exception)处理机制，在编译时会提示可能出现但未采取措施处理的异常情况，提醒程序员采取措施。

6、安全

Java 是应用于网络/分布环境下的语言，其安全性是极其重要的。因为 Java 代码可以在网络上发布和传递，若带有病毒，就会危害执行这些代码的所有客户(client)机器。因此 Java 的设计者充分考虑了安全问题，提供了多道关卡防止病毒的入侵。

Java 取消了指针，并对数组的下标进行检查，这样就防止非法改写系统内存。另外，对于与系统资源有关的对象，可以用 Java 语言的封装性来保护。Java 的设计者在编译后运行前增加了一次字节码检验，不管是从哪里得到的字节码，在它进入解释器之前，先由字节码检验器(bytecode verifier)检查它的安全性。Java 解释器决定各个类的存储器安排(类是 Java 的基本执行单元)，网络黑客(hacker)不可能了解某个类在内存的什么地方从而非法访问。类装载器(class loader)把从网络下载的各个类放入各自的存储空间，防止它们访问文件系统。所有这些考虑，使得 Java 对任何系统都很安全，完全胜任在网络上应用。

7、结构中立

Java 的结构中立性是要达到软件界一直追求的一个目标，即平台无关性，这也是当初开发 Java 的原因之一。

通过 Java 虚拟机(Virtual Machine)的指令代码实现了结构中立性。这种代码不能直接运行，但可以传送，由网络送到各种不同的计算机上，由该机上支持的 Java 解释器来执行。真正实现了一机编译，多机执行。这样看来：Java 的字节码很象一种计算机界的“世界语”，可在 Internet 上到处传送，然后由各地懂“世界语”的“翻译”成本地语言，让大家“听懂”。所以 Java 近乎完美地解决了平台无关性。

Java 的这种结构中立性不仅对网络十分有用，对软件开发也有很大意义。

8、可移植

除了采用虚拟机器代码的形式使 Java 语言程序可以在不同机器上运行外，Java 语言还采取其他措施提高可移植性。C 语言随着机器硬件和操作系统的变化而变化，这些变化使 C 语言程序的移植性大为减少。Java 语言则规定数据类型只有一种标准，不依赖于机器，避免了上述不可移植性。Java 各种类库也规定了可移植的界面，如 Window 类就能适应 Unix，Windows 和 Macintosh 环境。不但 Java 语言编写的应用程序可移植，连 Java 系统本身也具有可移植性。Java 的编译器是用 Java 语言实现的，运行系统用标准 C 实现，它们都有很好的移植性。

9、高性能

由于兼顾了可移植性、安全性、强壮性、结构中立等，其代价是 Java 的性能有可能降低，特别是解释执行字节码的速度显然比不上 C++ 执行机器码的速度。因此，Java 的设计者采取一些技术来保证高性能。其一是内建多线程，能提高 Java 程序的性能。其二是有效的字节码，编译后的字节码很接近机器码，可以在任何具体平台上有效地解释它。其三是在运行期间将字节码译成当地机器码，不过要花一定的延迟时间才能运行。其四是链接到本地的 C 语言代码，这样效率很高，但可能失去移植性。

10、多线程

所谓多线程是指在一个程序中可以同时执行一个以上的线程(thread)，也就是通常所说的并行执行多个任务。线程与进程(process)相似，也是执行中的程序，但线程数据较少，多个线程共享一组系统资源。

系统处理线程的负荷要比处理进程小。多线程的优点是可以合理调配多个任务，不同的线程处理不同的任务，之间互不干涉，不会由于一处等待而影响其他部分，这样就很容易实现网上的实时交互，具有实时特性。又如打印任务很花时间，如果程序是单线程的，

那么不打印完就不能做其他事情。采用多线程方法后。一个线程负责打印，另一线程继续干其他事，就不会浪费时间了。

11、动态

Java 的动态性是其面向对象设计的扩展。它提供运行时刻的扩展性，即在后期才建立各模块间的互连。各个库可以自由地增加新的方法和实例(instance)变量。这意味着现有的应用程序可以增加功能，只须链接封装有所需的方法的新类。

Java 用接口(interface)来实现多级继承，使用起来比 C++ 的多重继承更灵活。Java 语言的动态性使它能够胜任分布式系统环境下的应用，位于各地的类可以自由地升级，而不影响原 Java 应用程序的运行。

12、通用

Java 语言是通用的编程语言，利用应用编程接口 API(Application Programming Interface)扩充其编程方面的范围。图形处理 and 多媒体 API、网络 API、数据库接口 JDBC(Java DataBase Connectivity)API 等，使 Java 可以应用于网络、多媒体、数据库等领域。Sun 公司还不断推出新的 API 系列，进一步拓展 Java 的应用领域。

2.4 web 数据库技术

2.4.1 web 数据库介绍

2.4.1.1 Web 数据库概述

所谓数据库是长期储存在计算机内的、有组织的、共享的数据集合。关系数据库是我们最常见到的数据库。关系数据库最初设计为基于主机/终端方式的大型机上的应用，其应用范围较为有限。随着客户机/服务器方式的流行和应用向客户机方向的分解，关系数据库又经历了客户机/服务器时代。但是随着 Internet 应用的普及，由于 Internet 上信息资源的复杂性和不规范性，关系数据库在开发各种网上应用时显得力不从心。

Web 技术发展到今天，人们已经可以把 Web 技术引入到数据库系统中，于是 Web 数据库技术应运而生。数据库技术已经发展比较成熟，是计算机处理与存储数据的最有效、最成功的技术，适用于对大量的数据进行组织管理，而 Web 技术具有较佳的信息发布途径，这两种技术的天然互补性决定相互融合是其发展的必然趋势。将 Web 技术与数据库技术融合在一起，使数据库系统成为 Web 的重要有机组成部分，不仅可以把二者的所有优点集中在一起，而且能够充分利用大量已有的数据库信息资源，使用户在 Web 浏览器上方便地检索和浏览数据库的内容。

Web 数据库就是以后台数据库为基础，加上一定的前台程序，通过浏览器完成数据存储、查询等操作的系统。用户利用浏览器作为输入接口，输入所需要的数据，浏览器将这些数据送给后台服务器，由后台服务器对这些数据进行处理，例如，将数据存入数据库中，或者对数据库进行查询操作等，最后后台服务器将操作结果传回给浏览器，通过浏览器将

结果显示给用户。网站上的后台数据库就是一种 Web 数据库, Web 数据库可以实现方便、廉价的资源共享。

通常, Web 数据库的环境由硬件元素和软件元素组成。硬件元素包括 Web 服务器、客户机、数据库服务器、网络等;软件元素包括能够解释执行 HTML 代码的浏览器。在 Web 服务器中,必须具有能执行可以自动生成 HTML 代码的程序,如 JSP、ASP、PHP 等;具有能自动完成数据操作指令的数据库系统,如 SQL Server、MySQL、Oracle 等。

2.4.1.2 Web 数据库系统的体系结构

数据库系统的体系结构指的是带有数据库系统的计算机系统中各组成部分之间的相互关系,它是硬件、软件、算法、语言的综合性概念。研究数据库系统的体系结构就是要研究它的硬件分布和软件功能分配。从数据库系统开发的角度来看,只有站在系统的高度,以统一的体系结构作为指导,认真规划系统软硬件分布和功能分配,才能建立一个逻辑清楚、易于开发和维护的数据库系统。也就是说体系结构的正确设置和选择往往是整个数据库系统最终成功的关键。

八十年代以后,系统体系结构经历了从传统的二级客户/服务器(Client/Server, C/S)结构到三级 C/S 结构,直到今天以 Internet 为核心、以 Web 技术为基础的浏览器(Browser)作为软件集成环境,基于网络对象连接技术、多媒体文档结构以跨越多种操作系统和数据库平台的浏览器/服务器(Browser/Server, B/S)结构。C/S 结构在当时曾被称为新一代数据库技术,它适合于局域网环境运行,用户可根据需要在服务器上存取相应的数据,有一定的优势。随之 Web 应用的迅速发展,数据量日益增大,这种两层结构的不足就越来越明显。于是三层机构的 B/S/S(Browser/Web Server/DB Server)结构被提出,并受到广泛关注。

2.4.2 常用几种 Web 数据库比较

当前比较流行的 Web 数据库主要有 SQL Server、MySQL 和 Oracle 等。这三种数据库适应性强,性能优异,容易使用,在国内外得到了广泛的应用。

2.4.2.1 SQL Server 2000

Microsoft SQL Server 是微软公司从 Sysbase 获得基本部件的使用许可后开发出的一种关系型数据库,它建立在 Microsoft Windows NT 平台之上,提供强大的企业数据库管理功能。SQL Sever 2000 支持扩展标记语言(XML),具有强大、灵活的基于 Web 的分析,支持 OLE DB 和多种查询。SQLServer2000 是能满足用户需求,可扩充高性能的数据库管理系统。

由于出自微软之手,使得 SQL Server 和 Windows, IIS 等产品有着天然的联系。事实上以 Windows 为核心的几乎所有微软的软件产品都采用了一致的开发策略,包括界面技术、面向对象技术、组件技术等,这样在微软的软件中很多都可以相互调用,而且配合得非常密切。因此如果用户使用的是 Windows 操作系统,那么 SQL Server 应该是较佳的选择。

2.4.2.1 MySQL

MySQL 是当今 Unix 或 Linux 类服务器上广泛使用的 Web 数据库系统。它于 1996 年诞生于瑞典的 TCX 公司，支持大部分的操作系统平台。MySQL 的设计思想快捷、高效、实用。虽然它对 ANSI SQL 标准的支持并不完善，但支持所有常用的内容，完全可以胜任一般 Web 数据库的工作。由于它不支持事务处理，MySQL 的速度比一些商业数据库快 2-3 倍，并且 MySQL 还针对很多操作平台做了优化，完全支持多 CPU 系统的多线程方式。

在编程方面,MySQL 也提供了 C、C++、Java、Perl、Python 等 API 接口,而且有 MyODBC 接口,任何可以使用 ODBC 接口的语言都可以使用它。更重要的是,MySQL 的源代码是公开的,可以免费使用,这就使得 MySQL 成为许多中小型网站、个人网站的首选。

2.4.2.3 Oracle

Oracle 是 Oracle 公司开发出的一种面向网络计算机并支持对象关系模型的数据库产品。Oracle 支持大型数据库、多用户和高性能的事务处理。遵守数据库存取语言、操作系统、用户接口和网络通讯协议的工业标准，所以它是一个开放系统，保护了用户的投资。另外实施安全性控制和完整性控制，并支持分布式数据库和分布式处理。Oracle 是以高级结构化查询语言为基础的大型关系数据库，是目前最流行的客户/服务器体系结构的数据库之一。

以上介绍的三种数据库产品是目前 Web 上最常用的三种大型关系数据库系统，它们虽然在体系结构和操作方法上有许多相似的地方，但是在应用环境上还是各有侧重的。

表 2.5 列出了这三种数据库在性能和价格上的对比情况，在使用不同的系统时应针对实际情况采用合适的方案。

表 2.5 常用数据库性价比较

Tab2.5 The performance & price comparisons of commonly used databases

数据库系统	SQL Server	MySQL	Oracle
数据库费用	按连接计算，费用较高	全免费，较快	按计算机主频计算，费用较高，
操作系统及费用			
速度	价格较高快		价格较高，最快
容量	较大	较大	大
各份、恢复	好	较好	最好

从用户的技术水平以及国内软件应用的现状来看,SQL Server 应该是一个较好的选择。本文的系统采用 SQLServer2000 数据库系统进行开发。

2.4.3 Web 数据库的安全性

由于浏览器/Web 服务器/数据库服务器(Browser/Web Server/DBServer, B/S/S)模式的三层数据库结构的网络环境和自身的特点，使得 Web 数据库应用系统成为一个开放的系统，例如系统信息要在网络上传输，系统的功能全部基于网络实现，因此建设 Web 数据库系统比建设传统的数据库系统需要考虑更多的安全设置。采用 B/S/S 结构的系统所处的网络环境复杂，系统资源的分散，给系统的安全性带来许多不利因素，需要在多个层次上考虑系统的安全问题。并且从理论上来说，任何地方的计算机只要联网并安装浏览器就可以进入

数据库系统进行操作。正是这种方便性,为非法入侵提供了有利的条件。不过由于客户端不存放应用程序和数据,只要保证传到浏览器上的 HTML 源码不包含关键程序代码,客户端用户就不能看到更不能修改应用程序。各类服务器安全设计合理,基本可以保证应用程序和数据不被破坏。

一般数据库安全问题分为保证数据库系统中各种对象存取权的合法性(保证合法访问,阻止非法访问)和数据库内容本身的安全(防泄漏、篡改或破坏)两个方面。对数据库的安全可采用如下技术加以保证:

1、存取控制技术。一般采用多层控制,即系统登录控制、数据库使用权控制及数据库对象操作权控制。

2、隔离控制技术。通过某种中间机制,将用户存取对象隔离,用户不能直接对存取对象进行操作,而是通过中间机制间接访问,常用的中间机制有视图和存储过程等。

3、加密技术。加密技术是将数据库中的原始数据(明文)转换为人们所不能识别的数据(密文),以达到防止信息泄漏的目的。

4、信息流向控制技术。将数据库信息内容按敏感程度分成多少个级别(如绝密、机密、秘密、一般),以防止信息从高安全级流向低安全级。

5、推理控制技术。推理控制技术是防止用户从数据库合法访问到的信息中推断出他不了解的信息,推理控制技术特别是用于统计数据库的安全。

对于基于 B/S/S 体系结构的 Web 数据库系统,其安全性设计应从以下多个方面进行全面考虑:

1、硬件安全。系统硬件的安全性是系统正常运行的前提,包括硬件设备的安全设置和运行的可靠性。

2、网络安全。网络的安全要求网络操作系统及网络协议服务的正常运行,主要涉及到的网络安全关键技术有防火墙、加密技术、网络监控、代理服务器配置等。

3、服务器安全。基于 B/S/S 结构的 Web 数据库应用系统的资源都在各类服务器上,对这些资源的使用,应根据具体的情况,通过不同的用户赋予不同的权限,来对系统资源的访问内容加以限制。在用户使用服务器资源之前,首先应检查其合法性身份验证,然后在 Web 服务器上设计相应的权限限制,只有合法的用户才能根据自己的权限访问信息系统的网页。

4、应用程序安全。在业务处理程序中,应对各层次中的用户名及其口令加以屏蔽,决不能使它们以任何形式出现在客户端操作人员可以查看的地方,以免给非法入侵者以可乘之机。例如:客户端的浏览器上的 HTML 文件源代码中不能出现服务器的用户名和口令。

另外还要保证数据在信息传输中的安全,即 Internet 网络安全。整个信息系统的安全设计与 Internet 网络安全设计是密切相关的。Internet 网络安全主要包括:身份验证、授权控制、数据完整性、以及防止否认等。网络安全设置的目的是使系统的敏感性数据和机密信息受到保护,使之免受来自 Internet 用户的侵扰和攻击。

2.5 报表技术

信息为每一个企业提供了制定和实施成功的商业战略所需的原始素材。所以企业会在关键任务信息资产上投入数以百万美金的投资，如 ERP（企业资源规划）、CRM（客户关系管理）和 SCM（供应链管理），以捕捉、处理和存储与市场、客户、竞争对手和业务流程相关的有价值的信息。

然而这些系统只是实现成功所需要的解决方案的一部分。虽然这些系统能够高效的收集企业的运营信息，但是却通常并不针对不同信息消费者——如雇员、客户和供应商等提供分析、浏览和处理这些信息，并使其做出正确决策的功能。为了充分利用企业信息资产所带来的优势，每一个应用都需要一个可扩展的、可靠的、基于 Web 的解决方案，用于报表、分析和向相关的人员提供关键任务信息，以保证业务的成功。

关系型数据的报表工具 Crystal Reports 是具有强大内容创建和集成功能的高效的报表技术。Crystal Reports 被超过 300 家合作伙伴选为可视化的报表工具，代表着技术创新的领导者，迎接不断变化的报表开发和集成挑战。通过 Crystal Reports，可以快速的将任何数据转化为强大的、交互式的内容；将报表集成修改和查阅紧密集成到 .NET、Java 和 COM 应用中；使最终用户得以通过门户、无线设备和 Microsoft Office 文档对报表进行访问和交互。

全面的数据访问：Crystal Reports 中包括了超过 35 种数据源驱动，可访问任何一种 XML、OLAP、企业和关系型数据。您还可以通过 COM 数据驱动访问定制化的数据，并通过 JavaBean 获取更为灵活的数据连接性。

广泛的格式与设计：通过 Crystal Reports，您可以将数据转化为具有高度交互性的内容。您可以从 100 多种格式选项中进行选择，包括参数、映射、交叉表和超链接，用以强化您报表的冲击力。同时在 Crystal Reports 中还包括了超过 160 种公式和操作功能用以全面控制数据的展现。

强大而高效的工具：关键的报表要素包括了可存储在中央知识库（Repository）中的 SQL 命令、定制化功能、位图和文本对象，用于在多个报表中的重用。对 Repository 中对象的单点更新最小化了报表维护的要求。对于标准报表的快速格式化，可以创建并应用自定义的模板，甚至可将该模板用于已有的报表。

灵活的应用集成技术：Crystal Reports 带有灵活的 Java、COM 和 .NET SDK，以及可内嵌的 Report Application Server，可将报表与企业 Web 应用紧密集成在一起。通过使用 Crystal Reports 的零客户端浏览器控制，能够为最终用户提供丰富的报表交互、创建和修改功能。

报表的查看与交互：Crystal Reports 通过一系列的设备和环境实现对报表的查看与交互。使用 Report Parts 和对在 Office XP 中 Microsoft Smart Tag 的支持来提供通过门户、Microsoft Office 文档和无线设备对报表的访问、导航和交互功能。而先进的零客户端浏览器使最终用户能够对报表中的数据进行搜索，并导出为 Microsoft Excel、Word 和 HTML 格式，同时具有与源报表之间的超链接。此外，所有的报表都可以导出为多种不同格式，包括 XML、PDF、HTML 和 Microsoft Excel。

无限制的 SQL 控制：直接在数据库连接上编辑 SQL，实现无限制的控制 Unicode。支持在一个或多个报表中可以展现任何一种语言的数据定制化功能——从公式中提取业务逻辑，可在多个报表中重用 Repository——在集中的地点存储关键报表对象，可实现重用和单点更新。

Report Application Server：利用强大的基于服务器的报表处理服务将动态内容与企业 Web 应用紧密集成 Java、.NET 和 COM SDK——可采用任何一种开发平台来创建应用，并控制最终用户对报表的查看、交互和修改。

灵活的交付许可证：利用请求队列功能实现 Web 应用的灵活交付先进的零客户端浏览器——使最终用户能够搜索报表中的数据，并将结果导出为带有与源报表相关的动态链接 Report Parts——提供通过门户和无线设备对关键报表元素的访问向导性的导航--在一个或多个报表中的报表元素之间建立导航路径，使最终用户得以与报表进行交互。

公式组：在统一的地点实现公式和定制化功能的开发与维护。同时包括了无需编写代码的向导、查错和自动完成功能，实现更加轻松的创建。

导出——可导出为多种格式，包括 Microsoft Excel 和 PDF。产生的 Web 端报表如图 2.9 所示。



图 2.9 用 Crystal Report 产生的 Web 端报表

Fig.2.9 The web report of Crystal Report built

Crystal Reports 的 Java 支持有助于简化并加速访问、格式化，并将数据集成进 JavaServer Pages (JSP) 的过程。在 BEA WebLogic Workshop 和 Borland JBuilder 中，它是作为紧密集成特性提供的，并且属于默认安装的一部分。

第三章 系统分析与技术路线

3.1 研究目标

畜产品生产工艺流程是一项系统工程，是以科学的生产工艺将各个生产环节有机地联系起来，从而形成一套固定的生产模式，任何一个环节的疏漏都会影响生产线的运转，在实施过程中需要有高度的计划性和准确性。集约化养猪的目的是要摆脱分散的，传统的季节性的生产方式，建立工厂化、程序化、常年均衡的养猪生产体系，从而达到生产的高水平和经营的高效益。

在了解学习畜产品生产工艺的前提下，通过试验和构建商品猪精细养殖生产管理数字化平台一些关键技术部分，突出通用性，实现了商品猪精细养殖过程中如生产管理、饲料管理等环节的数字化，以满足现代化养殖的需要，提高工厂化养猪的质量及精度。

3.2 主要研究内容

以工厂化商品猪场为研究模型，经过实地调研，听取专家意见，抽象出一些共性，利用 Java 平台和 Web 开发语言，集信息处理分析技术、计算机网络技术、远程交互技术、网络数据库等技术，开发出一些通用性的构件和模块，逐步完善整个技术平台。对于商品猪养殖过程，有如下环节：

3.2.1 生产管理

工厂化养猪大部分采用先进的全进——全出流水式工艺。阶段饲养的养猪生产工艺由于设备、条件、规模的大小和分阶段划分的不同，饲养工艺也有差异。

所谓全进——全出是指在同一时间内将同一生长发育或繁殖阶段的工艺猪群，全部从一种猪舍转至另一种猪舍。流水式生产线是从猪的配种、妊娠、保育、生长肥育以至销售形成一条龙的流水作业，各阶段都有计划、有节奏地进行。如果繁殖节律以周为单位，那么每周都可以有计划地将一群母猪配种、一群母猪分娩、一群仔猪断奶、一群生长猪育肥，保证均衡生产不脱节，最大限度地利用猪舍设备。从猪出生开始一直到上市的整个过程，按母猪的同一生理阶段及其它猪的不同的生长时期，可划分为若干连续工艺阶段，每一阶段饲养着处于同一发育时期、具有同一饲养要求的猪群，经过一段饲养后，按工艺流程转到下一个阶段。工艺阶段间紧密结合，一环扣一环，均衡进行。由于设备条件、规模大小和分阶段的多少不同，其工艺流程也多样，目前这种饲养工艺大体有以下几种：

3.2.1.1 四阶段饲养三次转群工艺流程

四阶段饲养工艺是将种猪分成空怀和妊娠阶段，商品猪分成断奶仔猪阶段和生长肥育阶段。分别置于空怀妊娠猪舍、分娩哺乳猪舍（产房）、断奶仔猪培育舍和肥育猪舍内分区饲养。

四阶段流程：空怀妊娠母猪舍→分娩哺乳舍→仔猪保育舍→生长育肥舍。

（1）母猪空怀和妊娠阶段

这一阶段有空怀妊娠母猪分栏小群饲养的，每栏 4~6 头，也有空怀母猪单栏限位饲养，而妊娠母猪分栏小群饲养或空怀母猪分栏小群饲养，而妊娠母猪单栏限位饲养。这样配准的母猪在空怀待配区饲养 5 周，在妊娠母猪饲养区内饲养 5~11 周，然后转入下阶段饲养。

（2）母猪分娩哺乳阶段

同一周配种的母猪按预产期提前 1 周同批进入分娩母猪舍的分娩栏内，在此完成分娩产仔和哺乳。哺乳期为 4~5 周，母猪在这一阶段共饲养 5~6 周，断奶后，母猪回到空怀妊娠母猪舍参加下一繁殖周期发情配种。断奶仔猪则转入断奶仔猪培育舍饲养。

（3）断奶仔猪培育阶段

仔猪断奶后，同批转入断奶仔猪培育舍，在高床保育栏网上原窝或 2 窝仔猪小群饲养。在此饲养 5 周，体重达到 20kg 以上，再同批转入生长肥育猪舍肥育。

（4）生长肥育阶段

从保育舍转入到生长肥育舍的小猪，按肥育猪饲养要求，饲养 15~16 周，体重达 90~110kg 时同批出售上市。

四阶段饲养的优点是：猪群转群的次数相对少，减轻了转群的工作负担和猪只周转所造成的应激反应；工艺简单便于操作和控制；待配母猪、妊娠母猪和后备公猪在同一猪舍内分区饲养，减少了猪舍种类和猪舍维修；母猪产仔哺乳舍与仔猪培育舍相毗邻，便于调群和保暖设备统一安装。

3.2.1.2 五阶段饲养四次转群工艺流程

把空怀母猪和妊娠母猪编为一群，分娩哺乳母猪和仔猪为一群，仔猪断奶后进保育舍为一群，仔猪培育后转入育成舍为一群，最后为肥育群。五个阶段的猪群分别饲养在空怀妊娠母猪舍，分娩哺乳舍、仔猪保育舍、生长猪舍和肥育舍。

五阶段流程：空怀妊娠母猪舍→分娩哺乳舍→仔猪保育舍→生长舍→肥育舍。

五阶段饲养和四阶段饲养的不同之处，就是把商品猪再分为育成和肥育两个阶段，这样商品猪由四阶段饲养的仔猪和生长肥育两阶段变为五阶段饲养的仔猪、育成和肥育三个阶段、五阶段饲养可以保证猪只从断奶到上市的全过程中，最大限度地满足其生长发育的饲养营养、环境管理的不同需要，充分发挥其生长潜力，提高养猪效率。但与四阶段饲养相比，增加了一次转群的负担和猪只应激的机会。我国深圳广三养猪有限公司猪场的饲养工艺就是采用五阶段饲养，仔猪 28 日龄断奶体重达 7~8kg，转入断奶仔猪保育舍饲养 45 天，体重达 18~20kg，转入育成猪舍饲养 50 天，体重达 50~60kg，转入肥育猪舍饲养 60 天，体重达 90~110kg 出栏上市。深圳市光明合营猪场也是采用五阶段饲养，母猪饲

养阶段与四阶段饲养一样，仔猪 28 日龄断奶，从分娩猪舍转入断奶仔猪保育舍饲养 6 周，转入育成舍饲养 8 周后再转入肥育舍饲养 8 周，然后出栏上市。

3.2.1.3 六阶段五次转群的饲养工艺流程

在大型猪场，由于规模大，更便于实施全进——全出的流水式分阶段饲养工艺，划分的饲养阶段较多，专业分工较细。

六阶段流程：空怀母猪舍→妊娠母猪舍→分娩哺乳舍→仔猪保育舍→生长舍→肥育舍。

六阶段饲养工艺的特点是把空怀待配母猪与妊娠母猪分开，单独组群，利于配种，提高繁殖效率。而幼猪培育（70 日）以后，直接转入中猪生长阶段饲养，体重达 35kg 以后转入大猪肥育阶段。这种分阶段饲养的优点是：断奶母猪复膘快，发情集中，便于发情鉴定，易于掌握适时配种；猪只生长迅速，中猪生长阶段不因条件变化而生长受阻；便于猪群全进-全出，利于防疫保健。但六阶段的转群次数较多，增加了劳动量，同时还增加了猪只的应激反应。图 3.1 是常见猪场生产管理流程。

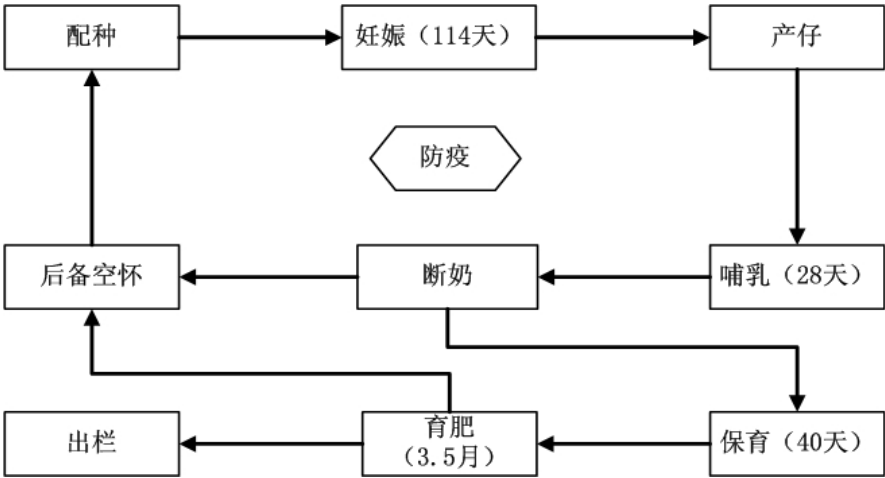


图 3.1 常见猪场生产管理流程

Fig.3.1The production management flow of common pig breeding farms

3.2.1.4 各阶段的管理工艺特点

工厂化猪场管理工艺需要要有较高的现代管理知识和组织技能，在母猪的配种、妊娠、分娩哺乳、幼猪保育、生长肥育等五个阶段中，最重要的是母猪的配种、分娩哺乳和幼猪保育这三个环节。

（1）配种母猪工厂化养猪要求全年保持均衡配种和产仔，按计划常年成批生产，保证每周有一定数量的母猪发情配种。如果配种头数不足或盲目配种头数过多，都会打乱整个生产流程，造成生产的混乱。因此，必须做到每周有一群母猪同期发情、同期配种、同期产仔对母猪的繁殖进行有效控制。

目前多数工厂化养猪场，常采用控制仔猪的断奶时间来达到母猪的同期发情。因为母猪的断奶时间有较大的变动范围，仔猪的断奶日龄可由 3 周龄到 6 周龄，这样就有可能使一组产仔相差 1~2 周的母猪在相同的时间内断奶。母猪断奶后一般在 3~7 天相继发情。

配种是工厂化养猪的重要的日常工作，必须认真做好。每周配种头数应视规模而定，如规模为 600 头基础母猪的猪场，每周应配种的母猪头数为 32 头，才能保证每周有 25 头母猪产仔。

(2) 妊娠母猪母猪配种后经过 21 天，如果不再有发情表现，可判断为已经妊娠，但配种后不再发情的母猪，并不一定都已妊娠。检查母猪是否妊娠的方法很多，而快速准确的方法是用超声波妊娠诊断仪，用于妊娠 4 周以上的母猪时，准确率可达 100%，这种仪器在妊娠期为 20 天后效果较好。现在进一步用超声波扫描投影方法，更早期、更直观可靠，准确率更高。

对已确定妊娠的母猪，将体重和年龄基本相近、配种时间一致的母猪编组置于限位单栏或小群同栏中饲养，以便掌握适宜投料量和猪的采食量。在配种 18~24 天和 38~44 天时要注意检查是否有重新发情的母猪，及时挑出返情母猪到配种舍配种。临产前 4 周和 2 周给母猪注射防仔猪下痢的疫苗及防制细小毒病的疫苗。临产前 1 周的妊娠母猪转入分娩哺乳母猪舍分娩栏内，以熟悉产房环境，作好分娩的准备工作。

(3) 分娩哺乳母猪临产前进入分娩栏的母猪，应给予严格清洗、消毒，避免病毒进入，分娩舍栏必须留有 1 周的时间维修设备和彻底清洗消毒，干燥后方可进猪。母猪临产时要有人在场，做好接产、断脐、断尾、打号、检查胎衣等工作，发现难产及时处理仔猪出生后 1 及时吃到初乳。

出生仔猪对寒冷十分敏感，做好保温工作对提高仔猪成活十分重要。分娩舍内温度一般保持在 22℃ 以上，为适应仔猪对温度的要求，应给仔猪创造小区温暖环境，增放仔猪保温箱，在保温箱上增设远红外板，或 60W 的红外灯泡，也可安放电热板。

在仔猪出生后几内就让它吃上初乳、吃好初乳，使之尽早获得免疫力，必要时人为的辅助仔猪固定乳头。

由于出生仔猪体内储铁甚少，从母体内获得也有限，又不能从饲料获得，且需要量很大，生后 3 天内如不及时补铁，那么 7~10 天就会出现营养性的贫血症。所以，仔猪出生后 3 天内注射血色素，或性血素 1 支 300mg，就可保证铁的需要量。

出生 3 天后就要训练仔猪饮水，7~10 天开始诱食补料，14~20 天让仔猪上饲槽采食乳猪料，以促使生长。

经常保持栏圈的清洁干燥，做到勤扫、勤清、圈内不积污水、尿液、粪便和残料。

(4) 保育阶段

仔猪断奶后转入保育阶段。这一阶段，仔猪与母猪不在一起，营养来源由吃奶供给转变为仔猪独立采食饲料。这种环境的变化，对仔猪是一个应激。因此，保育阶段的主要任务是创造条件，减少应激，缩短适应期，保持快速生长，防止拉痢掉膘。

保育舍实行高床小群饲养，仔猪原窝转群或两窝一群。保育的适宜温度和相对湿度控制在 20~22℃ 和 65%~70%，并注意良好的通风换气，保持圈舍清洁、干燥，饮水充足。

进入保育舍的幼猪，7~10 日内应保持原来的乳猪饲料，并严格控制采食量，由自由采食改为日喂三餐，投料量为自由采食的 70%。以后逐渐过渡到仔猪料。3~5 周龄断奶的仔猪，如下控制采食量，便容易诱发胃肠炎，造成增重减慢，甚至拉稀死亡。

保育阶段应安排驱虫、去势、防疫注射工作。

(5) 生长肥育阶段在进猪前, 生长肥育舍应进行维修和彻底地冲洗、消毒。进猪后保持舍内清洁、干燥、通风良好、饮水充足, 温度控制在 18~22℃, 夏季注意防暑降温。

转群时应将原圈猪按体重大小、性别、强弱分群, 每群大小应视圈舍大小而定, 一般为 10~20 头。

为防止争食和咬尾, 在小猪出生时应断尾, 在生长肥育圈内放些小铁环、小皮球、小石块之类硬东西, 让其耍戏, 分散精力, 增加运动量。如发现有咬尾现象, 应及时将咬尾者和被咬者同时调出, 分别单独饲养, 可避免全群咬尾发生。

肥育猪每月要定期称重, 以检查饲喂效果。经常检查猪群的采食、发育等情况, 及时调整饲料配方, 发现疫病及时报告, 采取有效措施进行治疗和处理。体重达 90~100kg 时应及时出栏上市。

3.2.2 饲料管理

3.2.2.1 配方管理

生长育肥猪的经济效益主要是通过生长速度、饲料利用率和瘦肉率来体现的, 因此, 要根据生长育肥猪的营养需要配制合理的日粮, 以最大限度地提高瘦肉率和肉料比。

动物为能而食, 一般情况下, 猪日采食能量越多, 日增重越快, 饲料利用率越高, 沉积脂肪也越多。但此时瘦肉率降低, 胴体品质变差。蛋白质的需要更为复杂, 为了获得最佳的肥育效果, 不仅要满足蛋白质量的需求, 还要考虑必须氨基酸之间的平衡和利用率。能量高使胴体品质降低, 而适宜的蛋白质能够改善猪胴体品质, 这就要求日粮具有适宜的能量蛋白比。由于猪是单胃杂食动物, 对饲料粗纤维的利用率很有限, 研究表明, 在一定条件下, 随饲料粗纤维水平的提高, 能量摄入量减少, 增重速度和饲料利用率降低。因此猪日粮粗纤维不宜过高, 肥育期应低于 8%。矿物质和维生素是猪正常生长和发育不可缺少的营养物质, 长期过量或不足, 将导致代谢紊乱, 轻者增重减慢, 严重的发生缺乏症或死亡。生长期为满足肌肉和骨骼的快速增长, 要求能量、蛋白质、钙和磷的水平较高, 饲料含消化能 12.97~13.97MJ/Kg, 粗蛋白水平为 16%~18%, 适宜的能量蛋白比为 188.28~217.57 粗蛋白克/兆焦 DE, 钙 0.50%~0.55%, 磷 0.41%~0.46%, 赖氨酸 0.56%~0.64%, 蛋氨酸+胱氨酸 0.37%~0.42%。肥育期要控制能量, 减少脂肪沉积, 饲料含消化能 12.30~12.97MJ/Kg, 粗蛋白水平为 13%~15%, 适宜的能量蛋白比为 188.28 粗蛋白质克/兆焦 DE, 钙 0.46%, 磷 0.37%, 赖氨酸 0.52%, 蛋氨酸+胱氨酸 0.28%。

3.2.2.2 配方计算

利用线性规划的优化方法设计畜禽优化饲料配方时, 如果取目标函数值为饲料成本最小化时, 则能有效地求出最低成本的最优配方。所谓的最低成本配方是在一组特定的约束条件下所获的结果。当这些约束条件包括原料的养分指标、预期达到的各种营养需要量、用量有限制的原料用量及原料价格等发生变化时, 一般会影响到最终的吨粮成本。

在原线性规划的配方模型上引入其对偶线性规划问题并求得参与优化计算的养分指标,能够从量上指导配方设计师如何有针对性调整饲养标准,达到快速有效降低饲料成本的目的。

一般情况下, 我们可将原始的线性规划的配方模型问题描述为:

(1) 目标函数为饲料成本最小化: $\min s = c_1 X_1 + c_2 X_2 + \dots + c_n X_n$

约束方程组:

$$\begin{cases} a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n \geq b_1 \\ a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n \geq b_2 \\ \dots \dots \dots \dots \dots \\ a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n \geq b_m \\ X_1 \geq 0, X_2 \geq 0, \dots, X_n \geq 0 \end{cases}$$

其中：决策变量 X_i ($i=1, 2, \dots, n$) 代表参与计算的原料用量； c_j ($j=1, 2, \dots, n$) 代表原料的价格； b_i ($i=1, 2, \dots, m$) 为期望达到的各种养分指标； a_{ij} ($i=1, 2, \dots, m$; $j=1, 2, \dots, n$) 代表不同原料含不同养分的数量，即养分系数。由模型 (1) 派生的线性规划问题如下：

(2) 目标函数为: $\max g = b_1 Y_1 + b_2 Y_2 + \dots + b_m Y_m$, 约束方程组转换为:

[illegible]

我们称(2)线性规划问题为(1)的对偶问题。在对偶线性规划问题中,决策变量为 Y_i ($i=1, 2, \dots, m$),为待求的资源 b_i ($i=1, 2, \dots, m$) 的影子价格或机会成本。问题(2)的养分序数矩阵是问题(1)的逆矩阵。

从运筹学理论上可以证明，问题（1）与问题（2）互为对偶问题；若原始问题和对偶问题都可行，则两者均有最优解，且两者的最优解相同。利用以上的对偶理论，在配方系统中增加了对偶模型的生成与应用模块，并将计算的参与优化计算的养分指标输出到结果中。

3.2.2.3 饲料库存管理

饲料的库存管理主要负责猪场饲料的入库、出库及库存的管理。入库、出库过程类似，入库是写入库单，修改库存；出库写出库单，修改库存。库存查询可以查出某种饲料的库存量。统计分析则可以设定多种指标进行统计分析，如不同饲养员领料情况、不同时间段饲料消耗情况、不同猪舍饲料消耗情况，还可以与饲养员的业绩挂钩等。

3.2.3 生产监测

生产监测子系统提出需要提醒生产人员或管理人员注意的一些信息，并提前报警，让生产人员或管理人员引起重视，从而避免因遗漏某些事项而造成的损失。

生产监测主要对以下信息进行监测并提供报警。

对于生产管理系统：提供对猪只转存的日期提醒，根据在系统初始化时在猪只存栏天数中设定的值结合猪只的进场日期，对猪只的每个阶段的转存予以报警，提醒管理人员及时将猪只转存；乳猪的早期断奶提醒；中猪的药物配方更改；大猪的停药提醒；母猪预产期提醒；母猪断奶时间提醒；配种时间提醒；空怀时间提醒；对饲料管理系统有饲料库存下限提醒；对疾病管理系统有药品库存下限提醒及重大疫情提示。

3.3 研究方法

3.3.1 技术路线

（1）统一风格与规范

为加快研究进度和减少重复劳动，本研究在充分分析现有相关专业软件的基础上，吸收成熟可行的软件设计思路，应用 Java 和 XML 语言并遵从 TCP/IP 协议进行程序设计。同时，以研究数字化技术平台为切入点，集中技术力量解决技术难点，制定出技术规范，形成统一风格。

（2）与领域专家密切合作

信息技术人员在了解用户需求基础上，多次请教领域专家，从专业角度提炼出研发系统应解决的技术难点，并提出技术平台的实施思路。在技术层面提出平台的软、硬件环境需求，提供养殖技术平台应用层面上的衔接问题，实现领域知识、系统开发与实施的相互衔接。

（3）标准化、组件化的设计方法

技术平台在满足技术指标的前提下，本着先进、可行和实用原则，进行总体设计。研究采用标准化、尤其是模块化设计，便于逐级分解、逐级集成。

（4）广泛采用成熟产品和技术的快速开发方式

信息技术的发展，使系统开发、建设的软、硬件装备费用大幅度降低，项目研究尽可能采用成熟技术和成型产品，加快研究和实施进度。

技术路线如图 3.2 所示。

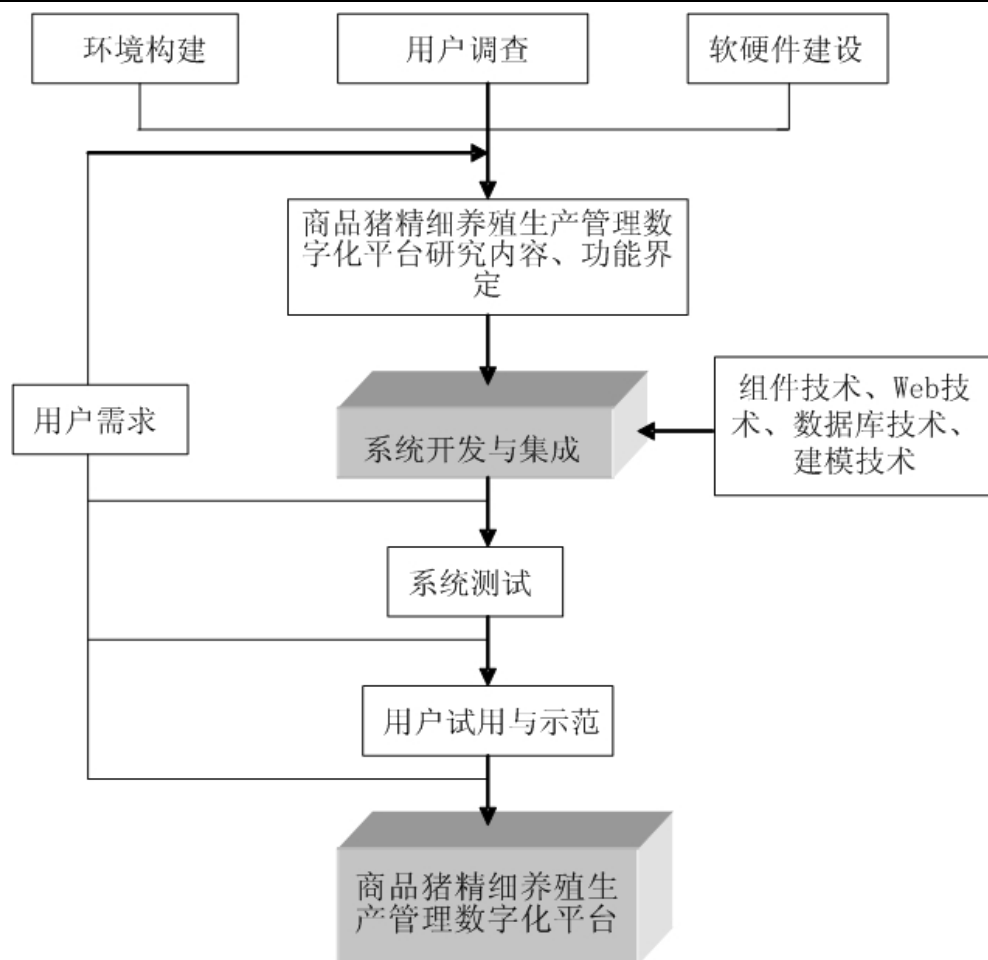


图 3.2 技术路线图

Fig.3.2 The figure of technical route

3.3.2 采用的系统架构

系统采用组件化的 B/S 架构充分实现系统的功能，并发挥用户的能动性，系统为用户提供方便友好的模型及知识库生成工具，实现了类似共享方式的共同开发，以无限地扩充、完善系统的可用资源。系统网络拓扑结构如图 3.3 所示。

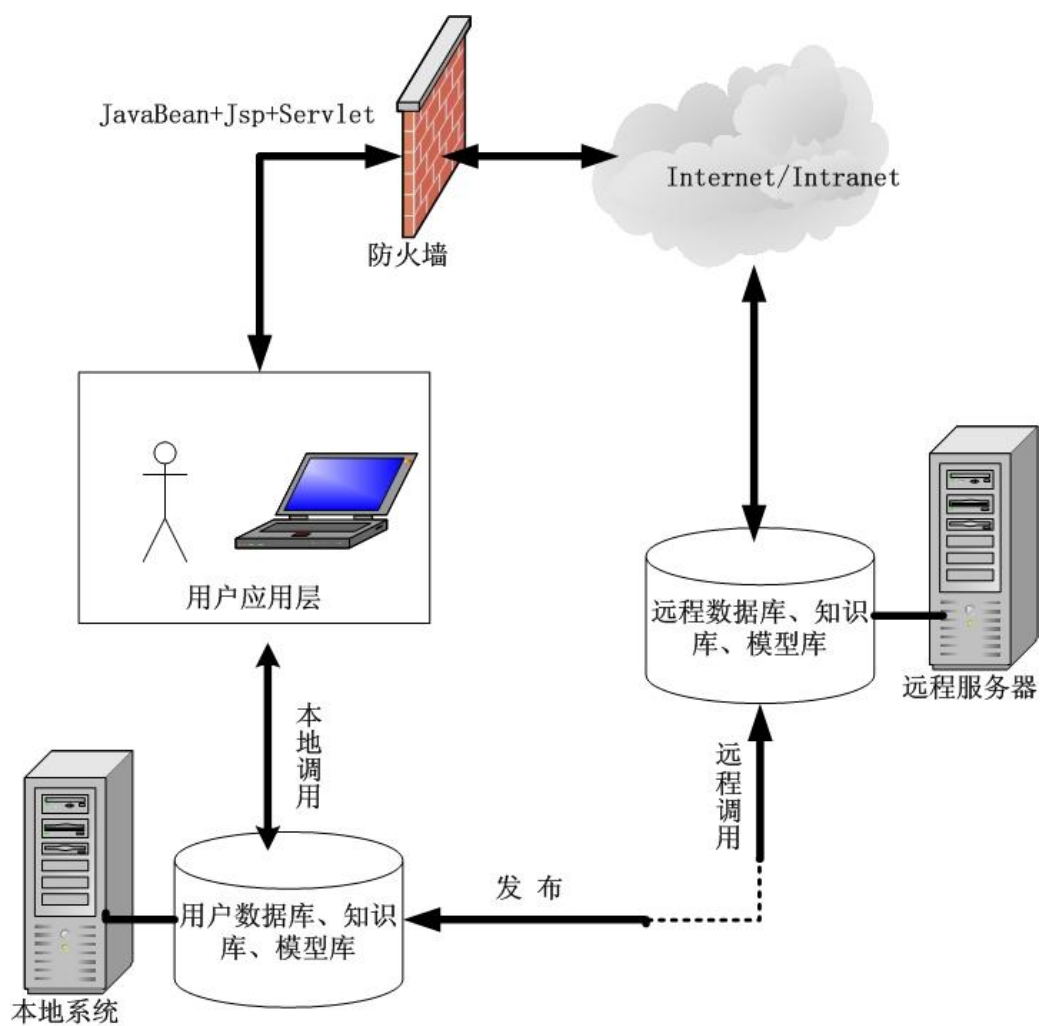


图 3.3 网络拓扑结构图

Fig.3.3 The figure of network topology

第四章 系统的设计与实现

4.1 系统总体设计

商品猪精细养殖生产管理数字化平台是商品猪精细养殖数字化技术平台的重要组成部分。商品猪精细养殖数字化技术平台从层次划分上可分为信息获取、信息处理和信息应用三部分，如图 4.1 所示。

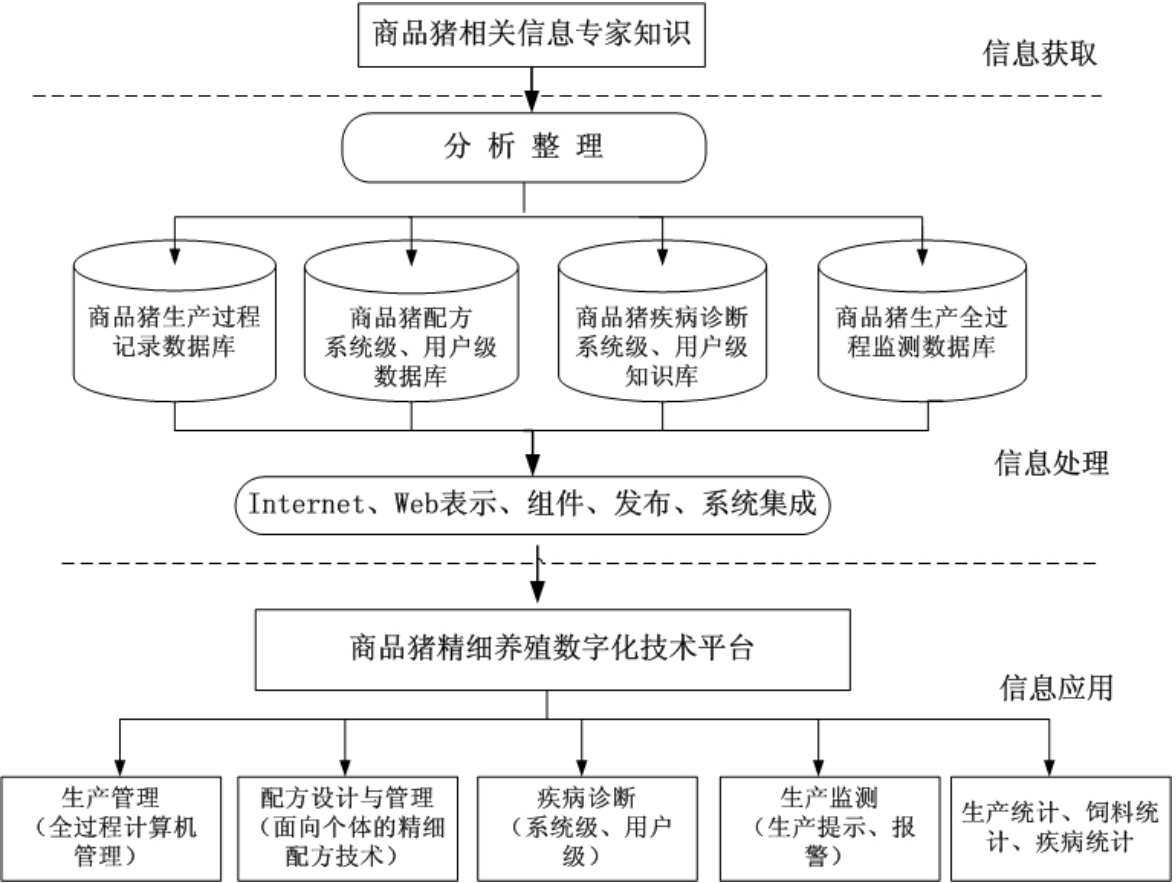


图 4.1 商品猪精细养殖数字化技术平台总体结构

Fig.4.1 The overall structure of the digitization platform for precision feeding of commercial pigs

在信息获取层面上，由系统分析和设计人员完成。包括前期的专业知识现场调研、专家走访、查阅资料自主学习等方式，获得最新的领域专业知识和第一手的数据材料。为接下来的系统分析设计，需求的提出做准备。

在信息处理层面上，由系统开发人员完成。在明确系统需求的前提下，建立商品猪生产过程记录数据库、商品猪饲料配方系统级数据库、商品猪疾病诊断系统级知识库和商品猪生产全过程监测数据库。在四大数据库后台的支撑下，运用 web 开发技术、组件技术、发布等技术手段在前端建立起基于 Internet 的面向广大用户群的 web 平台。

在信息应用层面上，面向分散在各地的广大用户。在整个数字化平台搭建好的基础上，他们可以通过界面友好、简便易行的 web 页面了解到与商品猪生产有关的专业知识和生产流程，并可以直接调用系统级的专业知识库去指导辅助实际生产。同时，不同的用户也可以根据自己地区、养殖场的实际需要，自行录入数据、添加知识到用户级数据库中去，指导以后的生产。

4.2 系统功能模块设计

4.2.1 总体功能模块划分

系统功能架构图如下：

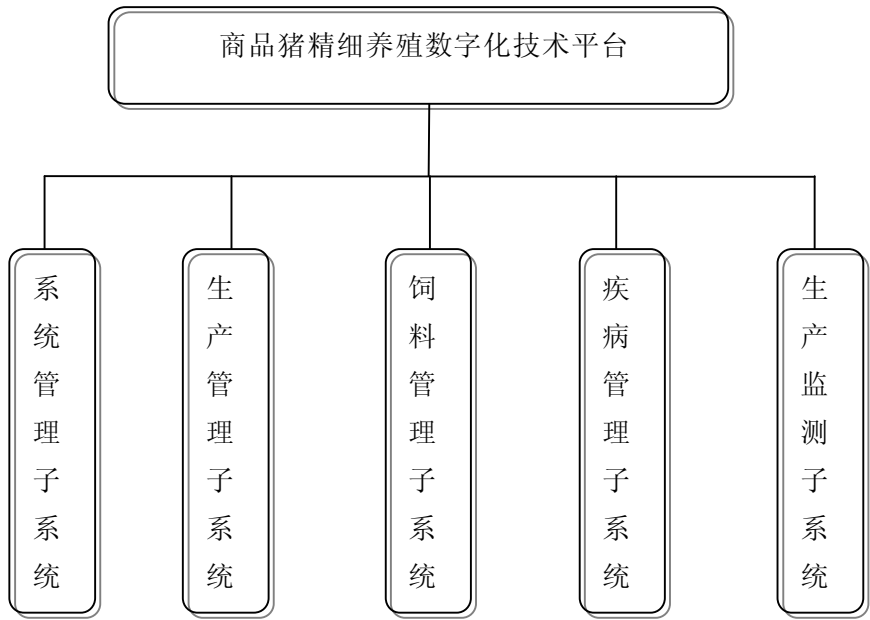


图 4.2 商品猪精细养殖平台系统功能架构图

Fig.4.2 The functional structure of the digitization platform for precsion feeding of commercial pigs

系统由五大功能子系统构成：

系统管理子系统：系统管理子系统主要负责系统参数设置、用户管理、用户维护信息、用户授权、系统访问记录及数据库操作记录查询、维护。

生产管理子系统：管理商品猪生产的全过程。包括乳猪、小猪、中猪、大猪的管理以及种公猪、种母猪的管理；猪的转存管理；饲料投放管理；饲料成本计算；饲料变更记录；猪只销售记录等功能。

饲料管理子系统：饲料管理子系统处理养殖场饲料、配方的相关问题。包括原料库管理、配方管理、饲料库存管理、配方计算。

疾病管理子系统：疾病管理子系统处理养殖场内猪的疾病相关问题。包括：疾病知识库管理、疾病知识库检索、药品信息管理、药品库存管理、疫情管理。

生产监测子系统：生产监测子系统负责上述所有模块中需要对生产管理人员进行提醒及预警的信息管理，生产人员可以在这个模块中看到所有的预期将要发生的情况。

4.2.2 所参与的子模块功能详述

本人独立开发的子模块有系统管理、生产管理和饲料管理三部分。

4.2.2.1 系统管理子系统

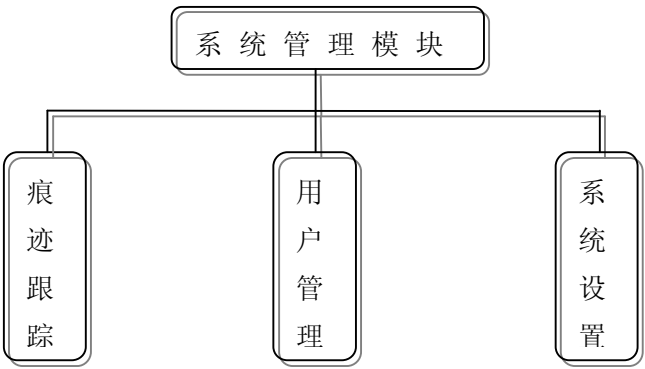


图 4.3 系统管理模块

Fig.4.3 The module of system administration

1、痕迹跟踪

用来记录每个用户登录系统后所进行的操作。主要对两种类型的操作进行记录：

(1) 页面访问记录：用户对所有页面的访问均被记录在页面访问记录表中，包括用户登录名、姓名、用户 IP 地址、访问时间、访问页面全路径信息。在此模块中，可以对用户页面访问记录进行组合查询，以反映用户对系统的访问情况。其用途是跟踪用户对系统内每个页面的访问情况，其结果用来对用户的访问进行统计评估、对有争议的操作进行取证、及对各子系统、子功能或单一页面的使用情况进行统计分析以评估业务流程的权重。统计分析结果可以用图形方式示出。另外，还可以按阶段对无须保存的记录进行条件删除。

权限：只限系统管理员操作。记录数据任何人员无权更改，包括系统管理员在内。

其流程图如下：

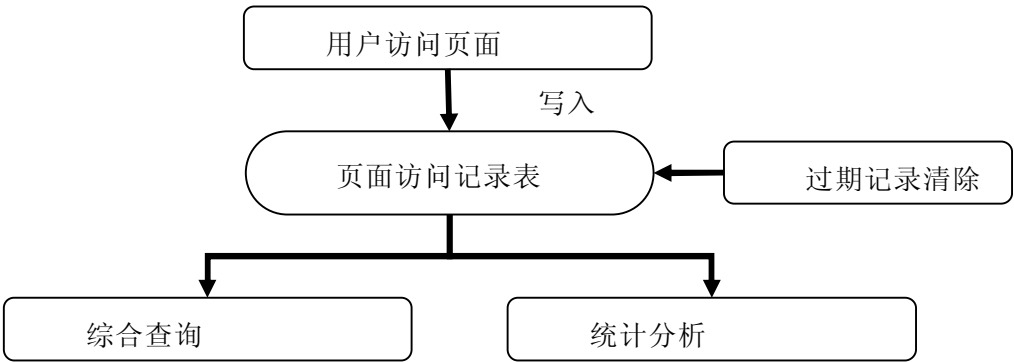


图 4.4 页面访问记录流程

Fig.4.4 The flow of the pages visited record

由此看出，页面访问记录功能应包含以下几个子功能模块以实现上述功能：

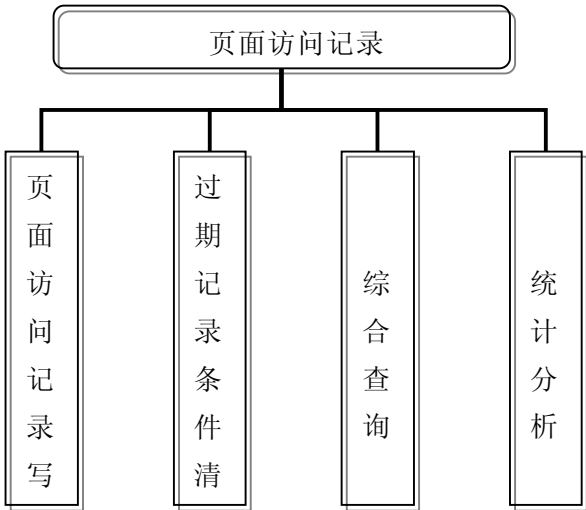


图 4.5 页面访问记录功能划分

Fig.4.5 The module of the pages visited record

（2）数据库访问记录：数据库访问记录，用户对所有数据库的访问均被记录在数据库访问记录表中，包括用户登录名、姓名、用户 IP 地址、操作时间、操作表名称、所进行的操作信息。在此模块中，可以对数据库操作记录进行组合查询，以反映用户对数据库的操作情况。其用途是跟踪用户对每个数据库的操作情况，其结果用来对用户的数据库操作进行统计评估、对有争议的数据库操作进行取证及对各数据表的使用情况进行统计分析以评估业务流程的权重。统计分析结果可以用图形方式示出。另外，还可以按阶段对无须保存的记录进行条件删除。

权限：只限系统管理员操作。记录数据任何人员无权更改，包括系统管理员在内。

流程如下：

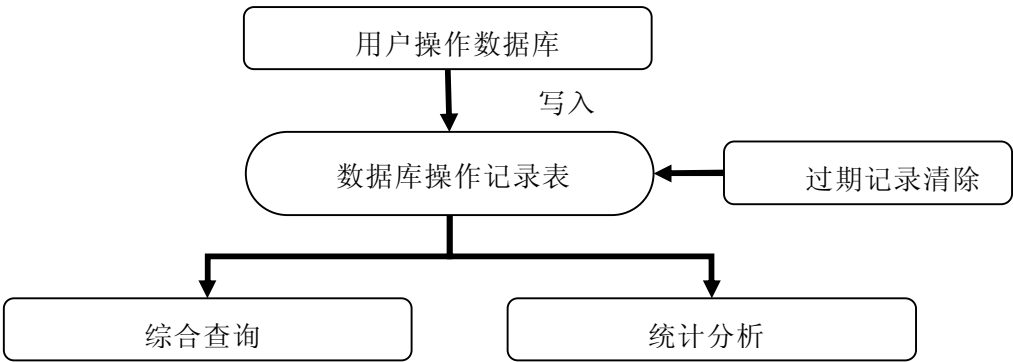


图 4.6 数据库访问记录流程

Fig.4.6 The flow of the database visited record

由此看出，页面访问记录功能应包含以下几个子功能模块以实现上述功能：

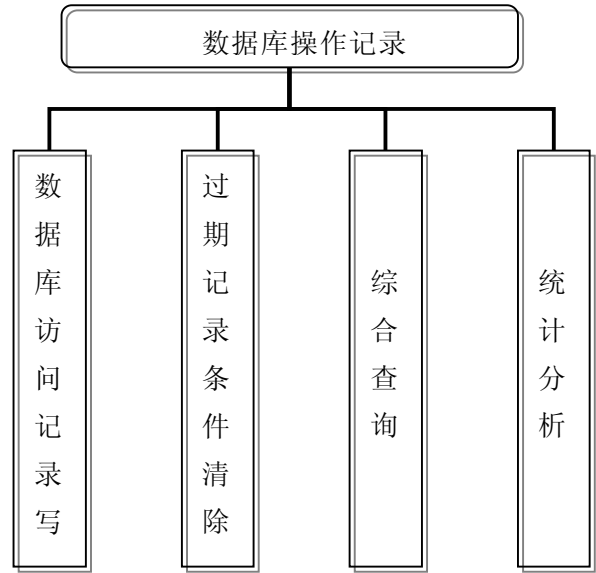


图 4.7 数据库操作记录功能划分

Fig.4.7 The module of the database visited record

2、用户管理

对用户信息进行管理：包括用户信息验证（即用户登录），用户首先完成系统登录后才能进入系统进行相应操作，否则用户无法使用系统；用户信息维护，包括添加新用户、删除用户、使用户无效、修改用户信息，修改用户密码、对用户完成授权操作功能。

（1）用户登录：用户登录过程是对用户录入的用户名和密码进行验证，这是用户操作系统的第一个步骤。用户名、密码验证无误，进入系统，并从用户信息表中取出相应数据（如用户名、姓名、单位、部门、权限、是否部门管理人员等）并置入 Session 变量，以便在整个会话过程中可以随时使用。否则停留在原登录页面上，等待用户输入正确的信息。这个过程与数据模型中的用户基本信息表和用户详细信息表对应，数据基本信息表中用户名做为主键，不得重复，同时做为数据详细信息表的外键，两表通过用户名字段内联接进行操作。如图 4.8。

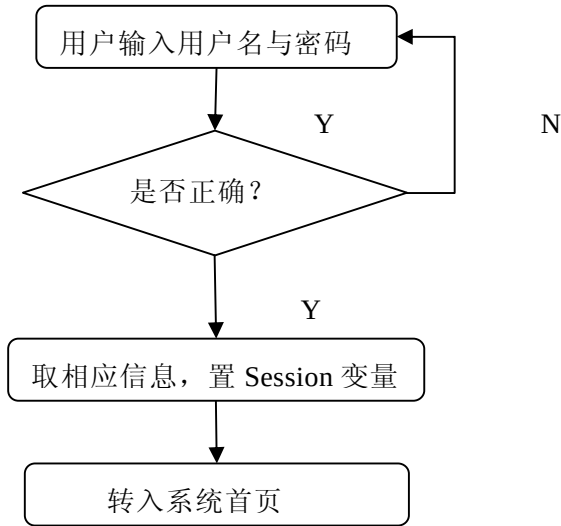


图 4.8 用户登陆流程

Fig.4.8 The flow of users' login

(2) 用户管理：用户管理模块管理用户的信息，对用户进行增删改操作，以及用户的授权及失效、生效操作。

权限：此模块除密码修改外，其他操作只限系统管理员操作。

此模块由以下几个子功能模块构成：

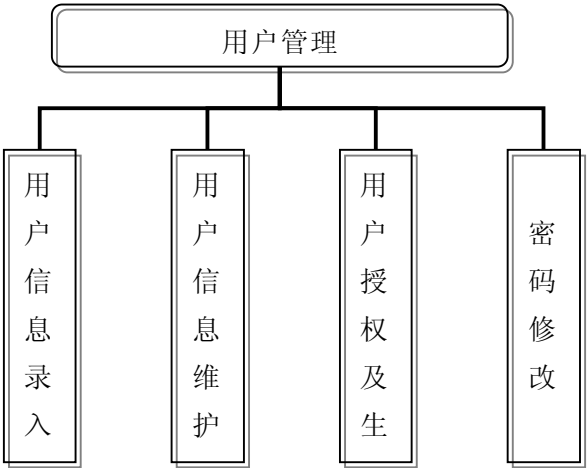


图 4.9 用户管理模块

Fig.4.9 The module of the users management

各子功能模块流程如下：

新用户信息录入：

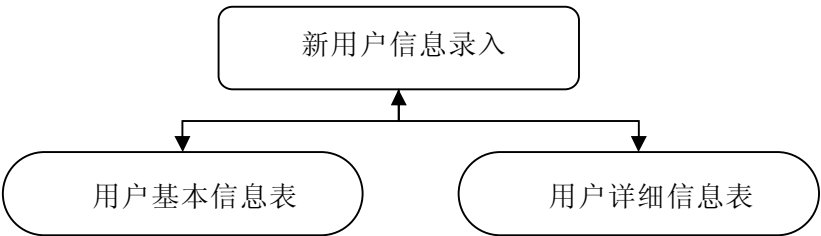


图 4.10 新用户信息录入流程

Fig.4.10 The flow of new users' information added

用户信息维护：

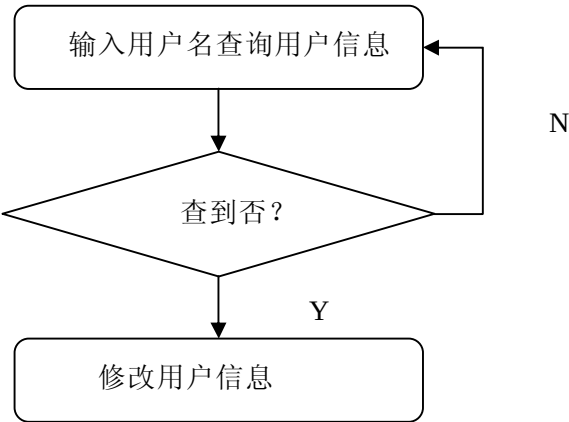


图 4.11 用户信息维护流程

Fig.4.11 The flow of new users' information managed

用户授权及生效:

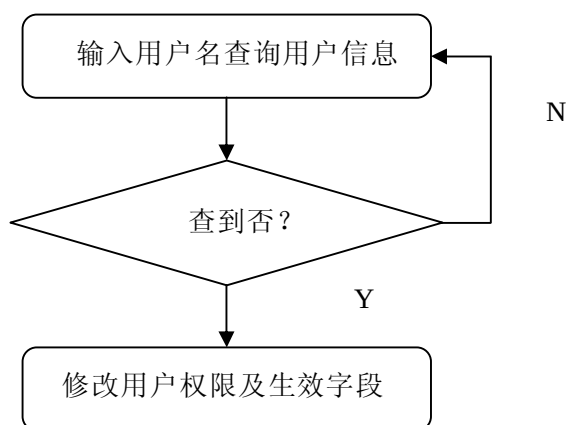


图 4.12 用户授权及生效流程

Fig.4.12 The flow of users' authorization

密码修改:

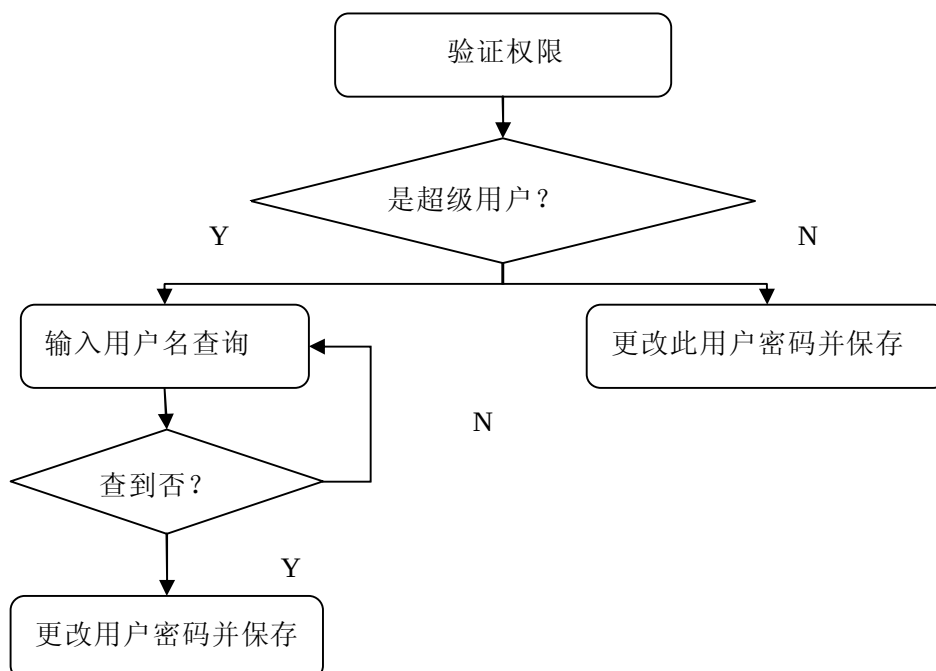


图 4.13 密码修改流程

Fig.4.13 The flow of users' password modified

3、系统设置：系统设置模块功能较为简单，它的功能是对系统内所有预先设置的表数据进行设置。提供对基础数据表增删改操作。

权限：只有系统管理员有权操作此功能模块。

需要进行设置的数据有：

岗位信息：猪场内的岗位设置信息。

职务信息：猪场内领导岗位职务信息。

单位名称代码：猪场内总场及分场的代码及名称信息。

部门名称代码：猪场内设置的职能部门的代码及名称信息。

职称系列：不同职称系列的代码及名称信息。

猪品种信息：猪的品种名称及代码信息。

存栏天数设定：包括妊娠天数、保育天数、网养天数、中猪天数、大猪天数、母猪断奶天数、配种间隔时间一、配种间隔时间二、停药时间、分娩时间参数的设定。（注意其中各参数单位不同，须在程序编码中注意计算及单位显示问题）

猪舍信息：各猪场内的猪舍信息。

猪舍性质信息：猪舍的性质名称及代码信息。

4.2.2.2 生产管理子系统

首先对猪场的整个生产流程详述如下：

猪场对猪的管理分为四个阶段：分别是乳猪、小猪、中猪和大猪，对于四个阶段的划分，一般猪场以天数来区分，专家告诉我们一般情况下，初生乳猪要在妊娠舍里呆 21-28 天，然后转到小猪舍，呆 42 天左右，而中猪和大猪的界限就不是很明显了。考虑到这个时间间隔可能会因为猪场的不同而不同，因此在系统设计中，我们将这些天数做成可以维护的，其功能在系统管理的参数设定里面对存栏天数表的维护。

针对上述情况，我们设计此子系统的功能模块构成如下：

生产管理子系统的一些约定如下：

在猪的转存、饲料管理中，系统管理的最小单位为猪舍，即管理一整栋猪舍内的猪只。对猪的进场，以及猪的销售管理直接管理到猪的个体，通过每个猪个体的耳号进行管理。

在生产管理过程中，对猪只的管理通过对猪舍的管理来实现。例如，我们系统中的约定为住在妊娠舍里面的都是乳猪，一旦转入到小猪舍，则其身份变为小猪；又如中猪住在中猪舍里面，但如果将其放到种公猪舍里，则它变为种公猪。

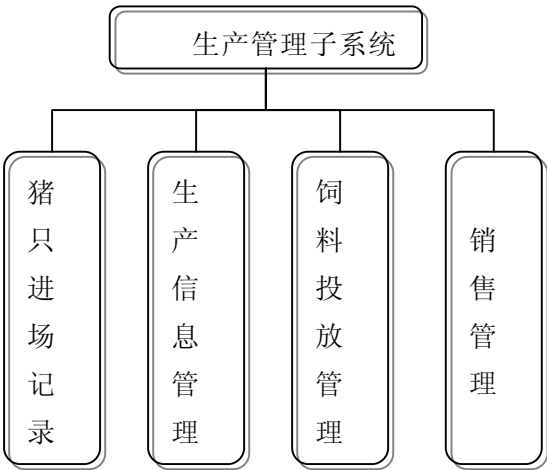


图 4.14 生产管理模块

Fig.4.14 Themodule of production management

1、猪只进场记录管理：这是整个生产流程的起点，在这一阶段，猪只出生或从其他地方通过某种途径（如购买，交换）进入猪场，获得自己在猪场内的唯一身份（一个号码，或打在耳朵上为耳号，这个号码将跟踪这只猪直到其出厂售出后仍然有效），并将其一些个体信息（如品种）记录在案。

权限：生产人员及管理人员。

流程如下：

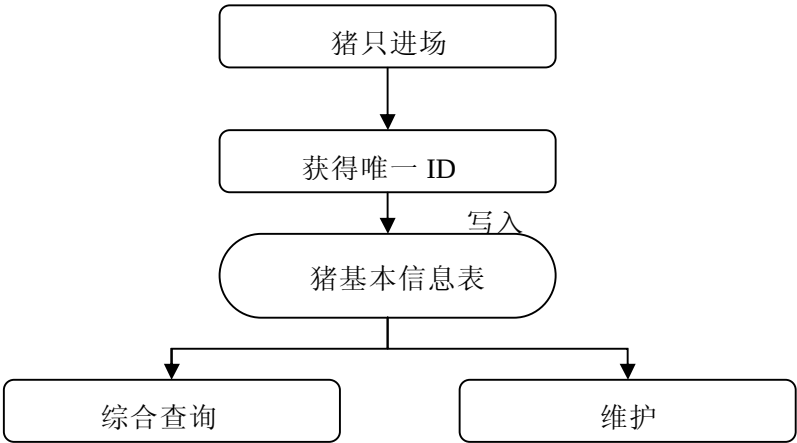


图 4.15 猪只进场记录管理流程

Fig.4.15 The flow of pigs-coming record management

2、生产信息管理：生产信息管理涉及到猪只进场后的一系列管理问题，包括：猪只的转存、当前猪只的信息查询、及转存的信息查询。因此，其模块构成如下：

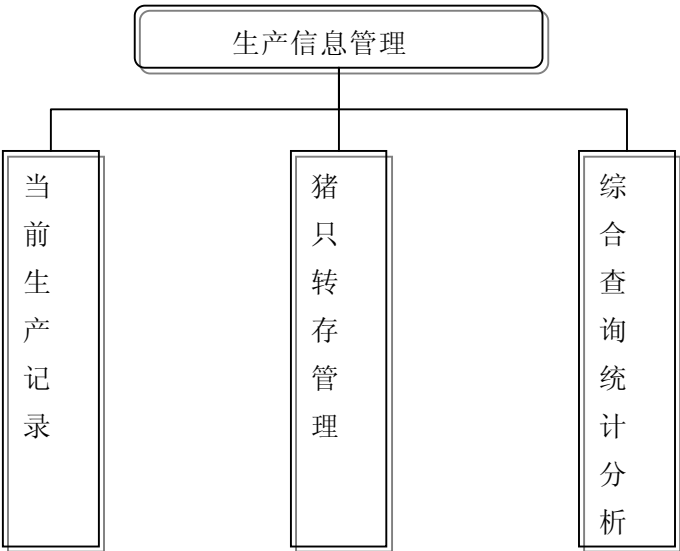


图 4.16 生产信息管理模块

Fig.4.16 Themodule of production information management

当前生产记录用来对猪只的当前情况进行查询，即查到的信息是猪只的最新情况，如猪只现在处于那个生长阶段，在那个猪舍里居住等信息。这些数据都写在猪生产记录表内。

猪只转存管理：猪只的转存管理较为复杂，在猪只的转存过程中，首先要完成猪只的转存即住址的变更，而后还要将这次变更记录在案。因此这个过程的详细描述如下：首先

更改猪生产记录表，将对应猪只的住址字段改为转存以后的住址，而后填写猪的位置变更信息表，写明这只猪由那个猪舍转到了那个猪舍，转存的时间，转存的原因等信息，换言之，猪位置变更信息表种的记录将完整跟踪猪只的户口信息。还有一种情况是，猪只被售出，这个情况更加复杂一些，当猪只售出，首先修改猪生产记录表，将其猪只字段改为空，将是否售出字段置“1”；然后写猪位置变更信息表，将目标猪舍字段置空，然后写猪只销售记录表，记录猪只的销售信息。其流程如下：

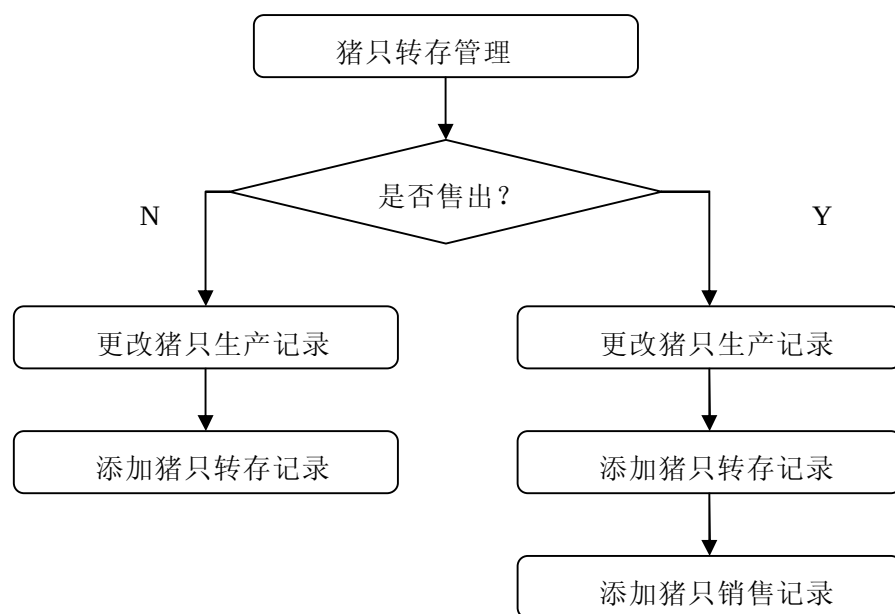


图 4.17 猪只转存管理流程

Fig.4.17 The flow of pigs-divering management

综合查询及统计分析：在猪生产记录表、猪位置变更信息表、猪只销售记录表的基础上，可以设计一些综合查询及统计分析功能，如根据猪只的存栏天数、品种、类型、等信息进行统计分析，并给出图文并茂的分析结果。

3、饲料投放管理：饲料的投放以猪舍为单位，这个功能模块记录每天每个猪舍的饲料投放种类及投放数量，并核算猪舍天饲料投放成本。这个模块与两个表有关：饲料信息表及饲料变更信息表。饲料信息表中记录猪舍每天的饲料投放量。饲料变更信息表中记录猪舍内饲料的变更及其原因。此模块还提供饲料的成本核算，需用到饲料管理子系统中的数据表中的数据；还提供一些统计分析功能，如饲料配方的投放量情况、猪舍的饲料投放曲线等。

权限：生产人员及生产管理人员

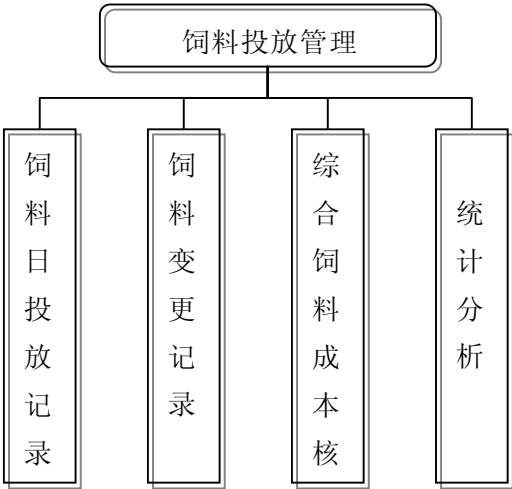


图 4.18 饲料投放管理模块

Fig.4.18 Themodule of feed put-in management

4、销售管理：销售管理主要记录猪只的销售情况，并跟踪猪只售出的第一层信息，即解决谁是买主的问题。在猪只转存的最后阶段，猪只被卖出，完成猪只转存记录中相应的工作后，转到此模块中完成猪只售出记录的填写。此转移过程应自动完成以保证流程的流畅。

4.2.2.3 饲料管理子系统

饲料管理子系统对猪场内的饲料及其相关问题进行管理。主要包括原料管理、配方管理及饲料的库存管理。

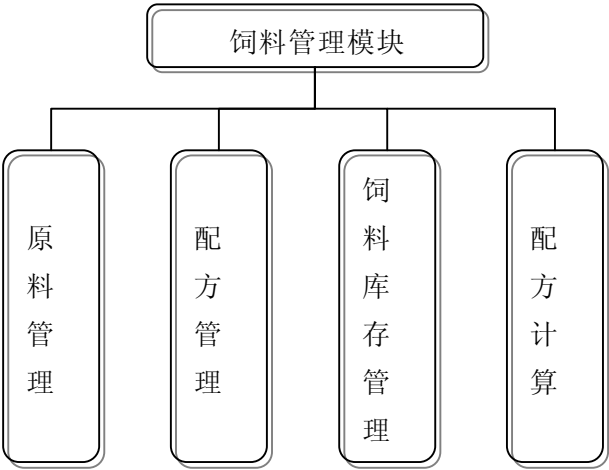


图 4.19 饲料管理模块

Fig.4.19 Themodule of feed management

1、原料管理：所谓原料管理，是指对饲料中的所含成分的管理，这个功能模块包括对原料信息表的增删改以及以图形化的方式显示饲料中的原料含量及营养成分含量等。

权限：饲养人员或管理人员

2、配方管理：配方管理对系统自带的配方及用户自定义配方进行管理。提供配方的录入、维护及查询功能。用户可以录入自定义的配方，也可将系统自带配方数据从库中调出

并在其基础上进行修改后形成独自の配方。原料库内没有的原料需先录入原料库而后再进入配方自定义模块定义配方。

权限：饲养人员或管理人员

模块构成如下：

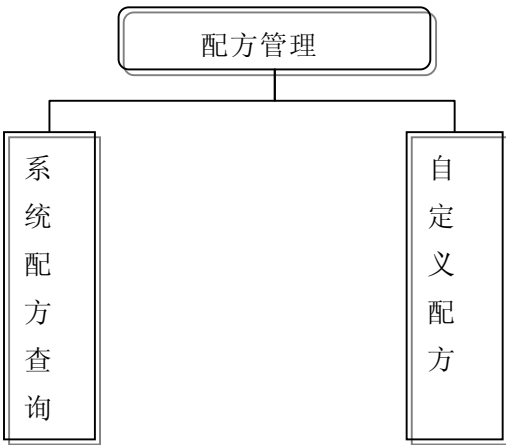


图 4.20 配方管理功能

Fig.4.20 Themodule of formula management

3. 饲料库存管理：饲料的库存管理主要负责猪场饲料的入库、出库及库存的管理。

权限：饲料库管及管理人员

管理流程如下：

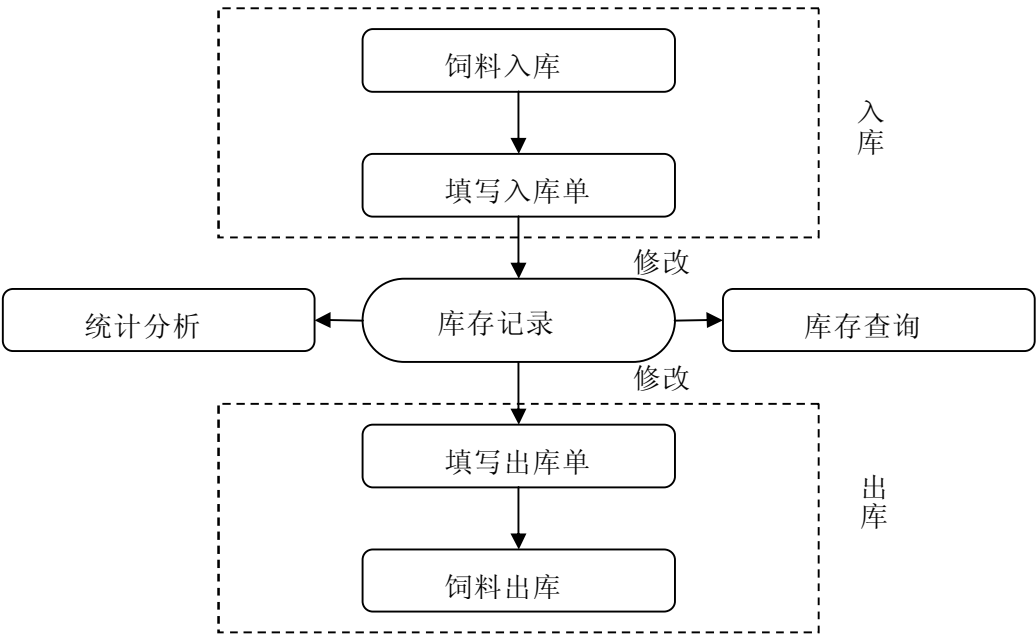


图 4.21 饲料库存管理流程图

Fig.4.21 The flow of feed stock information management

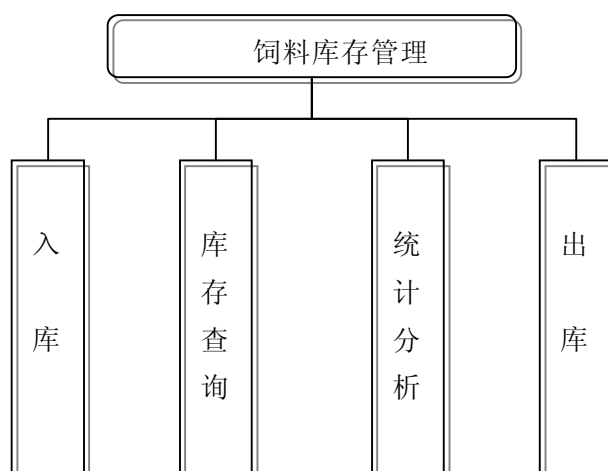


图 4.22 饲料库存管理功能图

Fig.4.22 Themodule of feed stock information management

入库、出库过程类似，入库是写入库单，修改库存；出库写出库单，修改库存。库存查询可以查出某种饲料的库存量。统计分析则可以设定多种指标进行统计分析，如不同饲养员领料情况、不同时间段饲料消耗情况、不同猪舍饲料消耗情况，还可以与饲养员的业绩挂钩等。

4. 配方计算：提供一种计算工具，利用线性规划的优化方法计算饲料配方，目标函数值为饲粮成本最小化时，求出最低成本的最优配方。

4.3 数据库结构设计

4.3.1 数据库设计的原则

数据库位于现代数据处理的中心，数据库的规划设计与实施是整个信息系统建设的核心。综合信息服务系统亦遵循这一原则，在系统设计中特别重视数据库设计。系统数据库设计是在 DBMS 的支持下，按照信息系统实际应用的要求，设计一个结构合理、使用方便、效率较高的数据库新系统。

根据系统实际情况的综合考虑，系统采用 B/S/S 数据库结构模式，数据库选用 SQL Server 2000，设计满足数据库第三范式。

在数据库设计中从整体上考虑以下几个方面：

1、用户权限分配

把系统用户划分为不同的类型，不同的用户赋予不同使用权限，用户登录后只能执行自己权限内的操作。

2、数据的有效性处理

用户在浏览器中填写的数据，在提交到数据库中之前，系统要对数据的有效性检测，如果填写不符合要求，会有相应的提示，确保输入的数据与数据库中的格式一致，从而保证数据库中数据的可靠性和完整性。

3、命名的格式要规范

不同的数据库产品对对象的命名有不同的要求，因此，数据库中的各种对象的命名、后台程序的代码编写应采用大小写敏感的形式，各种对象命名长度不要超过 30 个字符，这样便于应用系统适应不同的数据库。

4、数据库备份与恢复。再好的计算机也有可能出现故障，所以在数据库的管理过程中，数据备份是非常重要的一个环节。数据库的备份分为热备份和冷备份，热备份指在数据库运行时进行备份。对于 SQL Server2000 可以通过设置以让其在指定的时间内自动备份，冷备份指在系统运行一段时间之后，数据库管理员在数据库关闭时将数据备份到光盘或其它硬盘上。备份好的数据妥善保存，以便在数据库出现故障时及时恢复。

4.3.2 主要表结构

本系统数据库表总共 46 个，下面列出一些表结构：

表 4.1 用户基本信息表（YHJBXXB）
Tab4.1 The table of users' basic informations

名称	字段名	数据类型	字段长度	允许空	备注
用户名	YHJBXXB_YHM	VARCHAR	20	N	主键
密码	YHJBXXB_MM	VARCHAR	10	N	
用户权限	YHJBXXB_YHQX	CHAR	1	N	0—系统管理员 1—高层管理人员 2—部门负责人 3—用户
是否有效	YHJBXXB_SFYX	CHAR	1	N	0—无效 1—有效

此表为用户基本信息表，是用户登录、权限管理系统的核心数据模型，用户登录验证主要与此表打交道。其中用户 ID 为单一用户标识，用户名为用户系统登录名，密码为进入系统的身份验证密码，是否有效为用户有效标志；用户权限说明：系统管理员及高层管理人员对系统的访问不受任何限制；部门负责人对本部门业务相关模块拥有管理权限；一般用户拥有对系统的访问权限，已经对自己数据管理的权限。

表 4.2 用户详细信息表 (YHXXXB)

Tab4.2 The table of users' detail informations

名称	字段名	数据类型	字段长度	允许空	备注
用户名	YHXXXB_YHM	VARCHAR	20	N	外键
姓名	YHXXXB_XM	VARCHAR	50	N	
性别	YHXXXB_XB	CHAR	2	N	
出生日期	YHXXXB_CSRQ	DATETIME	8	N	
民族	YHXXXB_MZ	VARCHAR	20	N	
...	...	...	...	...	...
所属职称系 列	YHXXXB_SSZCXL	INT	1	Y	外键

此表记录所有用户的详细信息，并可以作为人事档案数据库、会计档案数据库及使用，同时承担一定的权限验证工作。

表 4.3 访问记录表 (FWJLB)

Tab4.3 The table of pages visited records

名称	字段名	数据类型	字段长度	允许空	备注
访问 ID	FWJLB_FWID	INT	4	N	主键（自动增加）
用户名	FWJLB_YHM	VARCHAR	20	N	外键
用户姓名	FWJLB_YHXM	VARCHAR	50	N	
登录时间	FWJLB_DLSJ	DATETIME	8	N	
登录 IP	FWJLB_DLIPDZ	VARCHAR	64	N	
访问页面名	FWJLB_FWYMC	VARCHAR	100	N	

此表由系统自动写入，用户对系统每次访问均记录在此表中，其中登录时间为用户开始访问此页面的时间，登录 IP 地址为用户机 IP 地址。

表 4.4 猪基本信息表 (ZJBXXB)

Tab4.4 The table of pigs' basic informations

名称	字段名	数据类型	字段长度	允许空	备注
猪编号	ZJBXXB_ZBH	INT	4	N	主键（自动增加）
品种编号	ZJBXXB_PZBH	INT	4	N	外键
性别	ZJBXXB_XB	CHAR	1	N	
入场日期	ZJBXXB_RCRQ	DATETIME	8	N	
入场重量	ZJBXXB_RCZL	NUMERIC	9	Y	
预计出场时间	ZJBXXB_YJCCSJ	DATETIME	8	Y	
来源	ZJBXXB_LY	CHAR	1	N	1—本场出生 2—购入 3—交换

表 4.5 猪舍信息表 (ZSXXB)

Tab4.5 The table of piggeries' informations

名称	字段名	数据类型	字段长度	允许空	备注
猪舍编号	ZSXXB_ZSBH	INT	4	N	主键 (自动增加)
场内编号	ZSXXB_CNBH	VARVCHAR	20	N	
场编号	ZSXXB_CBH	INT	4	N	外键
猪舍性质编号	ZSXXB_ZSXZBH	INT	4	N	外键
负责人代码	ZSXXB_FZRDM	VARCHAR	20	N	外键
备注	ZSXXB_BZ	TEXT	16	Y	

该表是确定猪具体信息的基础表, 根据猪舍信息可得到各种猪当前数量, 是确定一头猪或一群猪的依据。其中猪舍性质编号为猪舍性质表的外键, 通过猪舍信息, 可直接或间接得到猪所有的信息。

表 4.6 猪舍性质表 (ZSXZB)

Tab4.6 The table of piggeries' character

名称	字段名	数据类型	字段长度	允许空	备注
猪舍性质编号	ZSXZB_ZSXZBH	INT	4	N	主键 (自动增加)
性质名称	ZSXZB_ZSXZMC	VARCHAR	100	N	小猪、中猪、大猪、种公猪、种母猪等

本系统约定由猪舍性质可以确定猪只在生长过程中所处的阶段。

表 4.7 配种记录表 (PZJLB)

Tab4.7 The table of mating records

名称	字段名	数据类型	字段长度	允许空	备注
记录编号	PZJLB_JLBH	INT	4	N	主键 (自动增加)
猪编号	PZJLB_ZBH	INT	4	N	外键
所在猪舍	PZJLB_SZZS	INT	4	N	外键
发情时间	PZJLB_FQSJ	DATETIME	8	Y	
配种开始时间	PZJLB_PZKSSJ	DATETIME	8	Y	
二次配种时间	PZJLB_DECPZKSSJ	DATETIME	8	Y	
三次配种时间	PZJLB_DSCPZKSSJ	DATETIME	8	Y	
怀孕时间	PZJLB_QDHYSJ	DATETIME	8	Y	
分娩时间	PZJLB_FMSJ	DATETIME	8	Y	
断奶时间	PZJLB_DNSJ	DATETIME	8	Y	
空怀时间	PZJLB_KHSJ	DATETIME	8	Y	

4.4 系统开发平台的建立

本系统的开发是在 Windows2000 操作系统上开发的, 以 Tomcat5 作为 Web 服务器, 使用 JBuilder9+Dreamweaver2004 作为开发工具, 数据库采用 SQLServer2000 企业版(SP3), 业务逻辑封装在 JavaBean 和存储过程里, 结果显示用 JSP 和报表来实现。

JBuilder 开发工具是当前最优秀的 Java 程序开发之一, 它以强大的功能和友好的界面, 备受 Java 程序员的青睐。同时也可以开发 JSP 程序, 并有良好的编译和调试功能。DreamweaverMX 是 Macromedia 公司推出的主页编辑工具, 是一个所见即所得的网页编辑器, 在本系统中主要用于开发动态页面。

WEB 服务器采用 Tomcat, 自从 JSP 发布以后, 推出了各式各样的 JSP 引擎。作为一个开放源代码的软件, Jakarta-Tomcat 有着自己独特的优势。首先, 它容易得到, 完全免费。这也是很多人采用它作为 Web 服务器的原因, 任何人都可以从互联网上自由地下载。其次, 对于开发人员, 特别是 Java 研发人员, Tomcat 提供了全部的源代码, 包括 Servlet 引擎、JSP 引擎、HTTP 服务器——无论是对哪感兴趣的程序员, 都可以从这些由世界顶尖的程序员书写的代码中获得收益。最后, 由于源代码开放及世界上许多程序员的卓有成效的工作, Tomcat 已经可以和大部分的主流服务器一起工作, 而且是以相当高的效率一起工作。Tomcat 是 Apache Jakarta 的子项目之一。作为一个优秀的开放源代码的 web 应用服务器, Tomcat 基于 Java 技术, 全面支持 JSP2.0 以及 Servlet2.3 规范。因其技术先进、性能稳定, 而且免费, 因而得到了许多软件开发商的认可, 成为目前比较流行的 Web 应用服务器。

数据库工具选择在第二章已经讨论过, 选用方便易用并且性能较高的 SQLServer2000。系统体系结构如下图:

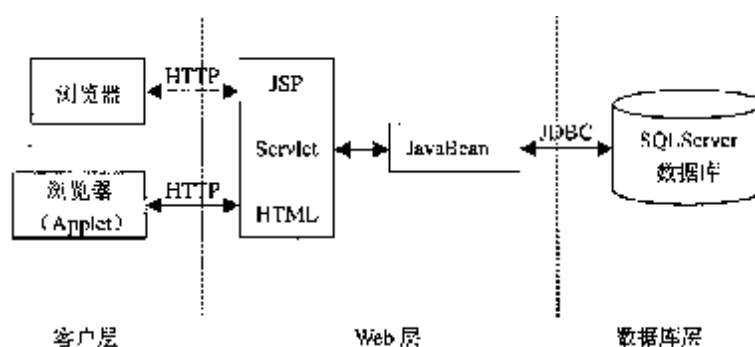


图 4.23 系统体系结构图

Fig.4.23 The figure of system structure

4.5 采用的关键技术

4.5.1 数据库优化技术

4.5.1.1 索引的使用原则

使用索引可以保证被索引列的唯一性和明显提高查询性能。但是，系统为维护索引，必将增加许多额外的开销，会降低插入、更新、删除操作的性能。所以，索引的使用是否恰当，对系统性能的影响很大。合理的使用索引，要注意以下两点：

表的规模不大时，不应该使用索引表需要频繁更新时，维护索引的开销大于使用索引减少的开销，所以这类表最好不要建立索引。

查询字段不确定的表不要使用索引，因为维护索引开销较大而不确定的查询字段使得索引并不能充分发挥作用。

4.5.1.2 数据的一致性和完整性

为了保证数据库的一致性和完整性，设计人员往往会设计过多的表间关联(Relation)，尽可能的降低数据的冗余。表间关联是一种强制性措施，建立后，对父表(Parent Table)和子表(Child Table)的插入、更新、删除操作均要占用系统的开销。如果数据冗余低，数据的完整性容易得到保证，但增加了表间连接查询的操作，为了提高系统的响应时间，合理的数据冗余也是必要的。使用规则(Rule)和约束(Check)来防止系统操作人员误输入造成数据的错误是设计人员的另一种常用手段，但是，不必要的规则和约束也会占用系统的开销，需要注意的是，约束对数据的有效性验证要比规则快。所有这些，在设计阶段应根据系统操作的类型、频度加以均衡考虑。

4.5.1.3 数据库性能的调整

在计算机硬件配置和网络设计确定的情况下，影响到应用系统性能的因素不外乎为数据库性能和客户端程序设计。大多数数据库设计员采用两步法进行数据库设计：首先进行逻辑设计，而后进行物理设计。数据库逻辑设计去除了所有冗余数据，提高了数据吞吐速度，保证了数据的完整性，清楚地表达数据元素之间的关系。而对于多表之间的关联查询(尤其是大数据表)时，其性能将会降低，同时也提高了客户端程序的编程难度，因此，物理设计需折衷考虑，根据业务规则，确定对关联表的数据量大小、数据项的访问频度，对此类数据表频繁的关联查询应适当提高数据冗余设计。

4.5.2 Web 技术

4.5.2.1 Session 技术

JSP 的 Session 对象可以记录每一个浏览器客户端的信息，Session 是个别独立存在的客户机信息，服务器通过这些信息来识别用户，从而提供不同的服务。JSP 的 Session 对象是一个内置对象，这个对象在第一个 JSP 页面被装载时自动创建，并被关联到 request 对象上。

在 JSP 程序里，每个客户端都对应一个 Session 对象来存放与这个客户端相关的访问信息。与 Application 对象存放的都是相同的公共信息所不同的是，Session 对象用来保存私人信息，每个用户各不相同。存放在 Session 对象内的数据，不仅可以是字符串，还可以是复杂的对象。Session 对象有它的存在期限，在 Tomcat 中，默认 Session 的存在时间为 30 分钟。如果超过 30 分钟没有向服务器提交进一步请求，就会被服务器从系统中清除。

可以使用 `setMaxInactiveInterval()`方法来设置 Session 存在的最大时间, 参数用秒作单位来计时。当用户退出时, 代表这个用户的 Session 对象将会被注销。

实际应用中, 用户登录后, 服务器就为该用户创建一个 Session 对象, 保存用户信息。当用户再访问以后的页面时, 可从 Session 中取出信息, 以便检查是否为合法用户, 从而实现系统访问控制, 保障系统的安全。以下是一个检查 Session 对象访问控制的实例:

```
<%
String sql session="Select YHJBXXB_YHM from YHJBXXB where
YHJBXXB_YHM='"+session.getAttribute("username")+"'", ;
//取出 Session 对象信息, 作为 SQL 语句检测的参数
ResultSet result session=data.executeQuery(sql session);//执行 SQL 语句
if (!result session.next){
response.sendRedirect("login.jsp");
//如果 Session 所存信息不合法, 则转入登录界面
}
else{//合法则允许用户使用相应的功能
}result-Session.close();//关闭数据集
%>
```

4.5.2.2 Web 服务器优化

Tomcat 是一个较理想的 JSP 和 Servlet 开发和支撑平台。但是 Tomcat 对 HTML 页面的执行效率比 Apache 要差, 因此考虑把 Tomcat 服务嵌入到 Apache 中, 把两种 Web 服务联合起来, 从而提高 Web 服务的效率, 加快浏览速度。具体的方法如下:

首先下载并安装 Apache 和 Tomcat。Apache 使用默认安装目录 C:\Program Files\Apache Group\Apache。Tomcat 使用默认安装目录为: C:\ProgramFiles\Tomcat。然后 <http://jakarta.apache.org> 去下载一个 mod_jk.dll 文件, 这是 Jakarta 组织开发的使 Apache 支持 Tomcat 的插件, 比 ApacheJserv 更强大, 因为 ApacheJserv 只能用在 Apache 上, 而不能用在别的 WebServer 上, mod_jk.dllr 可以和很多服务器搭配起来用。

4.5.3 MVC 模式的应用

MVC 是 Model/View/Control 的缩写。Model/View/Control 是软件设计的典型结构。如图 4.24 所示。

在这种设计结构下, 一个应用被分为三个部分: Model、View 和 Controller, 每个部分负责不同的功能。Model 是指对业务数据/信息的处理模块, 包括对业务数据的存取、加工、综合等; View 是指用户界面, 也就是面向用户的数据表示; Controller 则负责 View 和 Model 之间的流程控制, 也就是完成两个方向的动作: 将用户界面(View)的操作映射到具体的 Model, 以完成具体的业务逻辑; 将通过 Model 处理完的业务数据及时反应到用户界面(View)上。

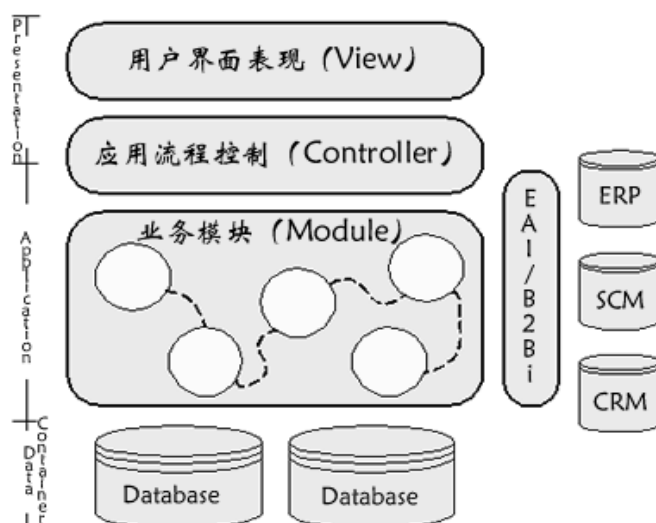


图 4.24 MVC 模式结构

Fig.4.24 The structure of MVC pattern

MVC 架构使得应用程序的结构更加清晰，通过将代码按照层次划分为业务逻辑/数据、用户界面和应用流程控制这三个层次，增强代码稳定性。对于 Model、View、Controller 这三部分功能来讲，View 的实现一般是由界面设计人员和界面程序员来完成，Model 则是由业务逻辑程序员来完成，Controller 则一般由负责整体控制的程序员来完成。这种模块功能的划分有利于在代码修改过程中进行模块的隔离，而不需要把具有不同功能的代码混杂在一起造成混乱。对于本系统开发而言，正是在项目小组内采用了 MVC 模式开发，将页面表示、程序流程控制和具体的业务逻辑分别归到三个层次上，层次清晰，层次间关系明确，有利于三个部分并行开发、加快项目进度。

4.5.4 组件技术应用

4.5.4.1 组件技术介绍

组件是具有一定功能的、能够独立工作或同其它组件组合起来协调工作的软件模块。每个组件包含一组属性、事件和方法，使用者通过组件的接口使用组件，而不关心其内部实现和运行方式。组件可以一次编写，随处执行，随处重用，只需要更新或替换相应组件就可完成对组件或程序的维护。

组件技术是指用可重用的组件来构造应用程序。软件的组件化不需要代码的重新编译和连接，而是直接将组件作为功能模块在二进制代码级用于软件指纹的装配。近几年来，组件在基于 WEB 应用开发中得到了广泛应用，应用组件技术可以明显提高开发质量和开发质量，极大的增强软件的重用能力，同时也使得程序的后期维护也变得极其方便。

JavaBean 是基于 Java 的组件标准，具有可重用、与平台无关、可移植、兼容性、可扩展、创建和使用方便的特点。JavaBean 组件可以在任何环境和平台上使用，满足各种交互式平台的需求，也可以在其它程序组件、WEB 应用中重用。在 WEB 应用中，通过 JavaBean 与 JSP、Servlet 的结合，可以极大的扩展程序的功能：将应用逻辑封装到 JavaBean 中，使

得用户界面与应用逻辑完全分离,能够更加方便的扩展应用逻辑,更加灵活的使用数据库;JavaBean 组件的重用提高了开发的效率和质量,维护也更加方便。

JavaBean 组件的主要设计目的如下:

紧凑而方便创建和使用;

完全的可移植性;

继承 Java 的强大功能;

支持可扩展的设计过程(设计环境, design-time)组件编辑器;

提高分布计算机制的强壮性。

通过对系统进行分析,在实际的操作过程中,系统需要完成大量的重复性的动作,比如数据库的插入、删除、修改等操作,会在各个不同的模块重复利用。采用面向对象和模块的设计思想,将各个模块抽象为类,充分体现了封装,继承和多态性的面向对象的思想。下次使用时只需要调用类即可,不但提高了开发的效率,也使得程序出错的可能性降低,提高了程序的质量,并且在程序代码的维护、扩展上也带来了很多方便。JavaBean 封装了所有业务逻辑部分,提供了有关数据库的操作,包括数据库的访问和维护,将复杂的功能划分到不同的层次来处理,体现了系统分层的思想。这种可重复使用和封装对提高动态 web 应用系统的开发效率以及改善系统的安全性都有十分重要的意义。

4.5.4.2 数据库组件

下面主要介绍一下有关访问数据库的类,封装了所有的数据库的访问和分页技术,并且根据不同的数据库结果集的类型,给出了不同的方法,例如有的数据库结果集支持滚动的字符集,有的结果集不支持滚动的字符集,对此分成两个函数来实现。这些操作几乎在每个模块中都会用到。在所有的数据库操作中,首先要用到数据库的连接,下面是取得数据库连接函数:

```
public Connection getConnection{Connection conn=null;
Statement s=null;
ResultSet rs=null;
Try{
conn=dbcon.getConnection("internal");
}catch (Exception e){
System.out.println(e); }
return conn;
```

根据 SQL 语句返回一个动态数组 Vector,数组的第一条记录存放对应记录字段的类型,由于 Vector 的容量可以动态增加,可以容纳任何数目的成员,因此采用 Vector 来存放查询到的结果集。下面就是此函数的具体的实现过程:

```
public Vector exec SqlWithResult(String sql){
.....
Vector v=new Vector;
try{
```

```

conn=this.getConnection();
s=conn.createStatement();
rs=s.executeQuery(sql);
int i=0;
while(rs.next()){
getData(rs,v,i);
i++;}
}catch(Exception e){ }
return v;
}}

```

4.5.4.3 分页组件

在数据库结果的显示当中，经常用到分页功能，因此把它写到封装到函数里面使用的时候只要调用此函数即可。传入三个参数，页号、每页记录、对应的 SQL 语句，返回一个动态数组 Vector，数组的第一记录存放对应记录字段的类型。此函数是针对支持滚动结果集的数据库实现的。

```

public Vector execSqlWithResult(int pageNo,int pageSize,String sql){
.....
Vector v=new vector();
LogPrintWriter dbLog=Logger.getInstance().dbPoolLog;
Try{
conn=this.getConnection();
s=conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
rs=s.executeQuery(sql);
rs.absolute((pageNo-1)*pageSize+1);
for(int i=0;i<pageSize&&!rs.isAfterLast(); i++){
getData(rs,v,i);
rs.next();
}
}catch(Exception e){ }
return v;
}}

```

下面这个函数也是实现分页功能，它和上面的不同之处在于数据库不支持滚动的结果集，不能采用绝对定位的方法取得记录，这里用一个变量 point 来保存当前数据库记录的位置，具体的实现步骤如下：

```

public Vector execSqlWithResultNotSupportScroll(int pageNo,int pageSize,String sql){
.....

```



```
Vector v=new Vector();
Try{
conn=dbcon.getConnection("internal");
s=conn.createStatementQ;
rs=s.executeQuery(sql);
int point=0;
while(rs.nextQ)
{if(point>=(pageNo)*pageSize)break;
else if(point>=(pageNo-1)*pageSize){
getData(rs,v,point-(pageNo-1)*pageSize);
point++;}
}catch {Exception e{}}
return v;}
```

只要在别的类中直接调用就可以了，可以使用如下调用方法：

```
public ExecuteSql executeSql=new ExecuteSQL;//执行数据库操作类
```

如果在 JSP 程序中直接调用，可以使用如下调用方法：

```
<jsp:useBean id="execSql" scope=" session" class="com.mcti.util.ExecuteSql"/>
```

4.5.4.4 字符集转换组件

因为 JSP 转成 Java 文件时的编码问题，默认的是 ISO-8859-1，如果一个 JSP 中直接输入了中文，JSP 把它当作 ISO8859-1 来处理，导致了在页面显示时出现了乱码。为解决这一问题，可以调用字符集转换组件解决，代码如下：

```
/**内码转换函数* @param GB
public static String GB2Uni(String GB)
{
    try {
        if (GB==null)
        {
            return "";
        }
        GB=GB.trim();
        byte[] tmp = GB.getBytes("Cp1252");//改变字节码
        String result = new String(tmp);
        return result;
    }
    catch(java.io.UnsupportedEncodingException e) {
        System.out.println("CommonFunction.GB2Uni
UnsupportedEncodingException"+e);
```

```

        return null;
    }
}

// 将 iso8859-1 编码转换为 gb2312 编码* @param str
public static String toGB(String uniStr){
    String gbStr = "";
    if(uniStr == null){
        uniStr = "";
    }
    try{
        byte[] tempByte = uniStr.getBytes("ISO8859-1");
        gbStr = new String(tempByte,"GB2312");
    }catch(Exception ex){
    }
    return gbStr;
}

```

4.5.5 其他关键代码举例

4.5.5.1 带输出参数的存储过程

本系统在猪只进场操作、饲料和药品进出库操作中大量应用了带输出参数的存储过程。以猪只进场为例，当进行进场操作时，首先要对猪基本信息表插入一条记录，由系统自动生成猪只的编号，同时还需要往猪只生产管理记录表和位置记录变更表写入该猪只进场的信息，其中就包括刚刚生成的猪只编号。这样，就用到了带输出参数的存储过程。存储过程代码如下：

```

CREATE procedure get_zjcbh
@zpz int, //输入参数：猪品种
@xb int, //输入参数：猪性别
@rcrq datetime, //输入参数：入场日期
@rczl numeric(9), //输入参数：入场重量
@yjccrq datetime, //输入参数：预计出场日期
@ly int, //输入参数：来源
@id int OUTPUT //输出参数：猪编号
as //插入一条记录到猪基本信息表
insert into zjbxxb
("zjbxxb_pzbh","zjbxxb_xb","zjbxxb_rcrq","zjbxxb_rczl","zjbxxb_yjccsj","zjbxxb_ly")
values(@zpz,@xb,@rcrq,@rczl,@yjccrq,@ly)
set @id=@@IDENTITY //设置输出参数

```

GO

用 JavaBean 封装了猪只进场的业务逻辑，函数 zjc_xx()中关键代码如下：

```
public int zjc_xx(String zpz,String xb,String ly,String zsbh,String rcrq,String yjccrq,String rczl){
```

```
    //以输入参数作为函数参数
```

```
    .....
```

```
    //执行存储过程，往猪基本信息表中插入一条新记录，并立即取出猪编号
```

```
        String str_pc="{call get_zjcbh(?,?,?,?)}"; //
```

```
        CallableStatement stmt=conn.prepareCall(str_pc); //预调函数
```

```
        stmt.setString(1,zpz); //设置输入参数
```

```
        stmt.setInt(2,Integer.parseInt(xb));
```

```
        stmt.setString(3,rcrq);
```

```
        stmt.setString(4,rczl);
```

```
        stmt.setString(5,yjccrq);
```

```
        stmt.setInt(6,Integer.parseInt(ly));
```

```
        stmt.registerOutParameter(7,java.sql.Types.INTEGER); //登记输出参数
```

```
        stmt.executeUpdate();//执行该存储过程并返回结果集
```

```
        id=stmt.getInt(7); //获取输出参数的值
```

```
    .....
```

在 JSP 页面中引入 zjc.class，调用其函数即可完成所有操作：

```
zjc.zjc_xx(zpz,xb,ly,zsbh,rcrq,yjccrq,rczl);//调用猪进场存储过程
```

4.5.5.2 JavaScript 的运用

在本系统中，大量运用了 JavaScript 实现一些 Web 页面中的功能和特效，通过 JS 文件封装起来，直接在页面中就可调用，方便易行。

输入为空的有效性验证示例代码如下：

```
function submit_onclick(bmxx)
```

```
{
```

```
    if( javaTrim(bmxx.BMMC)=="" ) //判断部门信息输入是否为空字符串
```

```
    {
```

```
        alert("请输入部门名称！"); //如果是空串，提示返回
```

```
        bmxx.BMMC.focus();
```

```
        return false;
```

```
    }
```

```
    else{
```

```
        return true;
```

```
    }
```

```
}
```

上面代码中, `javaTrim()` 也是一段 JavaScript 代码, 实现去掉字符串中的空格。另外, 系统功能树 `Xtree.js`, 图形化日期选取控件 `Calendar.js` 也在系统中反复大量使用, 只需在页面中写入:

`<SCRIPT language="javascript" SRC="../js/calender.js"></SCRIPT>` 语句, 给出调用路径, 然后在 `<input name="fqsj" type="text" id="fqsj" onclick="fPopCalendar(fqsj,fqsj);return true">` 中调用即可。

4.6 系统安全性设计

对于信息系统, 特别是基于 Web 的信息系统, 系统及其信息的安全性至关重要, 可以说没有系统及其信息的安全性, 就没有成功的信息系统应用, 因此安全性十分重要。

系统的安全性设计主要从操作系统安全、SQL Server 数据库服务器的安全和应用系统的安全三个方面进行的。对系统运行环境进行合理的安全配置, 防止非法人员攻击 Web 站点, 在编写应用程序时, 充分考虑安全性和容错性能, 以及在不同情况下应付不同行为的能力。

4.6.1 操作系统的安全

操作系统的安全是信息系统安全的基础, 数据库和应用程序的安全都是建立在操作系统安全之上的, 只有建立了安全的操作系统, 才能有安全的信息系统。本系统对操作系统的安全主要考虑以下几个方面:

- 1、用比较安全的文件系统, 如尽量使用 NTFS 格式的文件系统, 这种格式比 FAT 格式的文件系统要更安全。尽量将磁盘安装为只读模式, 仅有少数目录设置为读写状态。
- 2、废除系统默认的帐号和密码, 系统管理员的帐号和密码应及时改名, 同时尽量废止或者暂停 Guest 帐号的应用, 从而避免非法用户的攻击。还要限制用户尝试登录到系统的次数。
- 3、系统应注意及时保护和备份日志文件, 防止非法用户进入系统后抹掉其活动痕迹, 有利于查看和跟踪, 对重要目录也要及时备份。
- 4、关掉不必要的 TCP、IP 端口, 如提供 WWW 服务, 只需打开 80 端口, 以消除通过其它端口攻击系统的危险。
- 5、系统应及时更新, 以提高系统的安全性, 如 Windows 系列要及时打上最新的补丁。

4.6.2 SQLServer 数据库服务器的安全

安装 SQL Server2000 后首先打上补丁 sp3。然后再进行安全配置, 对于 SQL Server2000 主要考虑一下几个方面:

- 1、安全的帐号、密码策略。SQL Server 的认证模式有 Windows 身份认证和混合身份认证两种。如果数据库管理员不希望操作系统管理员通过操作系统登陆来接触数据库的话, 可以在帐号管理中把系统帐号删除。

SQL Server 不能更改 sa 用户名称,也不能删除这个超级用户,我们必须对这个帐号进行最强的保护,包括使用一个非常强壮的密码。最好不要在数据库应用中使用 sa 帐号,只有当没有其它方法登录到 SQL Server 实例(如当其它系统管理员不可用或忘记了密码)时才使用 sa。数据库管理员可以新建立一个拥有与 sa 一样权限的超级用户来管理数据库。很多主机使用数据库应用只是用来做查询、修改等简单的功能,可根据实际需要分配帐号,并赋予仅仅能够满足应用需求的权限。比如,只要查询功能的,那么就使用一个简单的 public。帐号能够使用 Select 功能就可以了。

数据库帐号的密码不能过于简单,如同系统密码不能过于简单一样,过于简单的密码容易被黑客获得。对于 sa 帐号更应该注意,不要让 sa 帐号的密码写于应用程序或者脚本中。另外要定期修改密码,数据库管理员应该定期查看是否有不符合密码要求的帐号。可以使用下面的 SQL 语句来查看:

```
Use master
```

```
Select name,Password from syslogins where password is null
```

2、加强数据库日志的记录。审核数据库登录事件的“失败和成功”,在实例属性中选择“安全性”,将其中的审核级别选定为全部,这样在数据库系统和操作系统日志里面,就详细记录了所有帐号的登录事件。还要定期查看日志,检查是否有可疑的登录事件发生。

3、使用协议加密。SQL Server 2000 使用的 Tabular Data Stream 协议来进行网络数据交换,如果不加密的话,所有的网络传输都是明文的,包括密码、数据库内容等等,都可能在网络中被截获,是一个很大的安全隐患。所以,在条件容许情况下,最好使用 SSL 来加密协议,但是这需要一个证书来支持。

4、对网络连接进行 IP 限制。SQL Server 2000 数据库系统本身没有提供网络连接的安全解决办法,但是 Windows 2000 提供了这样的安全机制。使用操作系统自己的 IPSec 可以实现 IP 数据包的安全性。对 IP 连接进行限制,保证只有自己的 IP 能够访问,拒绝其他 IP 进行的端口连接,对来自网络上的安全威胁进行有效的控制。

5、禁止别人探测 TCP/IP 端口。默认情况下,SQL Server 使用 1433 端口监听,即使在更改之后,也可能被人通过微软未公开的 1434 端口的 UDP 很容易探测到 SQL Server 使用了什么 TCP/IP 端口。因此,必须首先在实例属性中选择 TCP/IP 协议的属性,选择隐藏 SQL Server 实例。对 Windows 2000 操作系统来说,在 IPSec。过滤拒绝掉 1434 端口的 UDP 通讯,尽可能地隐藏 SQL Servers 这样就禁止对试图枚举网络上现有的 SQL Server 实例的客户端所发出的广播作出响应,别人就不能用 1434 来探测 TCP/IP 端口了。然后在实例属性中选择网络配置中的 TCP/IP 协议的属性,将 TCP/IP 使用的默认端口变为其他端口,从而确保数据库的安全。

4.6.3 应用系统的安全

应用系统的安全在这里主要指应用程序源代码的安全。虽然 JSP 程序运行于服务器端,执行后只会返回给用户标准 HTML 代码,但还是存在源代码暴露的问题,所以要尽可能防止源代码以明文方式返回给访问者,确保系统的安全。

1、采用 **JavaBean** 封装系统的事务处理逻辑。系统对数据库访问的操作、中文字符转换、系统时间的相关操作等都采用一个或多个 **JavaBean** 封装起来，这样不仅提高了代码的可重用性和系统的开发效率，更重要的是增加了黑客对系统的入侵和攻击难度，也增加了系统可用于弥补安全漏洞和消除隐患的时间，以便在黑客进行破坏之前解决好自己的安全问题，从而增强系统的安全性。

2、**JavaBean** 的存放安全。在 **JSP** 应用系统中，当前目录下都有一个 **WEB-INF** 目录，通常用来存放 **JavaBean** 的编译后的文件 ***.class**，这个文件中一般都存放了访问数据库的用户名和密码。如果不给这个目录设置合理的权限，**class** 文件就可能泄漏出去，从而给系统安全带来极大隐患。通过设置 **Apache** 和 **Tomcat** 服务器环境可以解决这个问题。首先将 **class** 文件的存放的默认路径进行修改，并且把该目录设置为只能执行而不允许读取，即在 **httpd.conf** 文件中添加放置 **class** 的目录并设置 **Deny from all** 等属性。

3、用户访问限制。用户使用系统之前，系统首先获取用户的 **IP** 地址，然后将该 **IP** 地址与系统预先设置好的用户 **IP** 地址进行比较。如果 **IP** 不满足系统要求，则拒绝该用户访问。**JSP** 中用于获取的 **IP** 地址的方法是：

```
ip=request.getRemoteAddr();
```

如果 **IP** 地址合法才能进入用户登录界面。系统还对不同用户赋予不同的权限，不同的用户登录之后访问的权限也不一样。并且限制用户登录的次数，例如，可设置连续输入三次错误则拒绝访问。在验证用户名和密码之前，还要对用户输入表单信息进行检测，过滤一些类似引号、等号和分号等特殊字符，防止破坏者构造恶意的 **SQL** 语句非法登录。

4、删除示范脚本。**Apache** 和 **Tomcat** 服务器都带有相当数量的示范脚本，而这些脚本都具有强大的功能，可能被用来进行恶意破坏。因此在系统正式使用之前，要将 **Apache** 和 **Tomcat** 服务器中所有的示范脚本全部删除掉，进一步加强系统的安全性。

经过以上对系统的安全设置，可以让系统本身具备足够的安全防范能力。当然，更主要的还是要加强内部的安全控制和管理员的安全培训，而且安全性问题是一个长期的解决过程，还需要以后进行更多的安全维护。

4.7 系统实现

4.7.1 系统运行软件环境及配置

本系统所选用的服务器运行环境为：

操作系统：Windows2000 Server（SP4）

Java 运行环境：J2SDK 1.4.2

Web 容器：Tomcat 5.0.19

数据库：SQLServer2000（SP3）企业版

报表服务器：Crystal Enterprise Report Application Server 9.2（SP2）

客户端要求支持及兼容 **IE** 的浏览器，推荐使用 **IE6**（SP1）。

4.7.1.1 Java 运行环境配置

从 <http://java.sun.com/j2se/1.4.2/download.html> 下载 j2sdk, 安装完 j2sdk 以后, 在我的电脑->属性->高级->环境变量->系统变量中添加以下环境变量(假定你的 j2sdk 安装在 c:\j2sdk1.4.2):

```
JAVA_HOME=c:\j2sdk1.4.2;
classpath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
path= %JAVA_HOME%\bin;
然后可以写一个简单的 java 程序来测试 J2SDK 是否已安装成功:
public class Myjava{
public static void main(String args[]){
System.out.println("试试看, 装好了吗? ");
}
}
```

将上面的这段程序保存为文件名为 Myjava.java 的文件。

然后打开命令提示符窗口, 进入 Myjava.java 所在目录, 然后键入下面的命令

```
javac Myjava.java
java Myjava
```

此时如果看到结果的话表示安装成功, 出错则仔细检查一下你的配置情况。

4.7.1.2 Tomcat 运行环境配置

从 <http://www.apache.org/dist/jakarta/> 下载 tomcat5.0.19, 安装 Tomcat 后, 在我的电脑->属性->高级->环境变量->系统变量中添加以下环境变量(假定你的 tomcat 安装在 c:\tomcat5):

```
CATALINA_HOME=c:\tomcat5;
CATALINA_BASE=c:\tomcat5;
```

然后修改环境变量中的 classpath, 把 tomat 安装目录下的 common\lib 下的 servlet-api.jar 追加到 classpath 中去, 修改后的 classpath 如下:

```
classpath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;%CATALINA_HOME%\common\lib\servlet-api.jar; 接着可以启动 tomcat, 在 IE 中访问 http://localhost:8080, 如果看到 tomcat 的欢迎页面的话说明安装成功了。
```

4.7.1.3 SQLServer2000 的 JDBC 驱动配置

安装好 SQLServer2000 企业版, 打上 sp3 补丁包后。到 SQLServer2000 的官方网站上下载 jdbc driver, 找到 jdbc 的安装目录, 把 lib 目录下面的 msbase.jar 和 mssqlserver.jar、msutil.jar 三个文件一起 copy 到 \$CATALINA_HOME/common/lib/, 然后在系统的环境变量 classpath 加入:

```
%CATALINA_HOME%\common\lib\msbase.jar
%CATALINA_HOME%\common\lib\mssqlserver.jar
```

%CATALINA_HOME%\common\lib\ msutil.jar

保存后即可正常使用 JDBC 驱动。

4.7.1.4 报表服务器配置

要使 Tomcat5 与水晶企业报表应用服务器 CE RAS 9.2 协同运行，需要按以下步骤配置：

(1) 将 Program Files\Common Files\Crystal Decisions.0\crystalreportviewers 文件夹复制到 Tomcat 5\webapps\ROOT 文件夹下。

(2) 修改 webapps\ROOT\WEB-INF 文件夹下 web.xml 文件，加入 context 参数来制定 crystalreportviewers 文件夹的位置：

```
<context-param>
<param-name>crystal_image_uri</param-name>
<param-value>/crystalreportviewers</param-value>
</context-param>
```

(3) 在 Tomcat 的 webapps\ROOT\WEB-INF 文件夹下创建 lib 文件夹，然后将 Program Files\Common Files\Crystal Decisions.0\jars 文件夹下的所有 JAR 文件复制到该文件夹。

(4) 重新启 Tomcat 服务器即可完成配置。

4.7.2 系统运行

在浏览器地址栏中键入地址，如 <http://localhost:8080/pig-login.jsp>，进入到系统登陆页面，如图 4.25：



图 4.25 系统登陆页面

Fig.4.25 The login page of system

输入正确的用户名密码后，进入到系统托盘页面，如图 4.26：



图 4.26 系统托盘页面

Fig.4.26 The menu page of system

从上图可以链接到平台的六大系统中去。点击系统管理，进入到该子系统的主页面，然后选择系统设置，如图 4.27：



图 4.27 系统设置页面

Fig.4.27 The page of system setting

该模块提供对系统人员参数、猪只参数、猪舍参数等进行初始化。从系统托盘进入到生产管理系统，出现页面如图 4.28：



图 4.28 生产管理页面

Fig.4.28 The page of production management

生产管理模块针对猪生产流程各阶段进行控制，提供对猪只进场信息记录、生产信息、饲料投放和销售的管理。从系统托盘中进入到饲料管理系统，出现页面如图 4.29：



图 4.29 饲料管理页面

Fig.4.29 The page of feed management

饲料管理子系统对猪场内的饲料及其相关问题进行管理。主要包括原料管理、配方管理、饲料的库存管理及配方辅助计算。从系统托盘可以进入到统计系统，该子系统集成了生产管理、饲料管理和疾病管理中的所有报表和统计图，如图 4.30 所示：



图 4.30 猪只进场统计

Fig.4.30 The page of pig-coming statistic

图 4.31 对整个猪场内所有猪只种类数量进行了统计，上方以不同颜色的柱状图表示，下方用报表的形式列出。



图 4.31 原料库存统计

Fig.4.31 The page of feed stock statistic

图 4.31 对原料库存种类数量进行了统计汇总，并用蓝色提示超出库存上限，红色提示低于库存下限，对应进行出库、入库操作。

第五章 系统应用案例

本系统适用于具有一定规模的工厂化养猪场、养殖企业，为其生产过程提供辅助支持。

5.1 工厂化养猪的条件和优势

工厂化养猪必须具备一定的条件，才能发挥工厂化养猪的优越性，以达到高产、优质、低耗、高效的生产目的。工厂化养猪必须具备以下几个条件：

- 1、选用优良猪种或配套品系猪，建立猪的良种繁育体系，选育优良品种，筛选最佳杂交组合。
- 2、采用先进的繁育技术，提早断奶、同期配种，实行流水式作业和“全进全出”生产。
- 3、根据猪的不同生理阶段对营养的需求配合全价配合饲料，实行标准化饲养。对生长肥育猪采用直线饲养，才能实现产品规格化。
- 4、设计适合当地气候条件的猪舍，为猪群提供适宜的生长和繁殖环境，还要有利于卫生防疫和环境保护。
- 5、具备配套的机电设备：如供水、供料、供暖、保温、通风、降温 and 污水处理设备，重点搞好母猪分娩舍的设备和仔猪培育的环境。
- 6、严格执行科学的兽医卫生防疫措施，有效地预防和控制传染病和普通病的发生。
- 7、实行现代化企业的经营管理，推行计划管理和经济责任制，建成产供销一体化的生产系统。
- 8、工厂化养猪工艺的实施，要求猪场具备一定的规模，只要有了相当的规模，才能发挥其技术优势和取得好的经济效益。

工厂化养猪的优越性表现在以下几个方面：

1、能大量提供商品猪肉

工厂化猪场规模大，采用科学养猪方法，肥猪出栏快，商品肉量多。以 1984 年扩建的广东省佛山市圩岗畜牧场为例，目前年生产瘦肉型猪已达 3 万头，按佛山调市城区 50 万人口计，五个 3 万头猪场即可以解决每人每年供应 24 公斤猪肉的需要量。年产 15 万头肉猪则每天可上市 411 头，肉猪每头重 100 公斤，产肉量按 80% 计。每天可上市猪肉 32889 公斤。

2、提高定额，节省劳力，降低劳动强度

在人力养猪的条件下，一个人最多也只能饲养 250~300 头肉猪或 20~30 头母猪，而且工作量大，劳动繁重。从美国引进三德畜牧设备公司生产的万头养猪生产线，存栏母猪 500 头，公猪 20 头，肉猪 4500~4700 头，在美国为 3 人管理，在我国为 7 人管理。平均一个人一年可以生产肉猪 1300 头。工厂化猪场由于供水、供料、冲洗猪栏均用机械操作，工人职责主要为观察猪的采食、发情、配种和接产，检查健康，栏内辅助清扫，控制及检查机

器和调整猪群、称重记录等，若以一人一年养 1000 头猪计，一人一年的劳动量为 2920 小时，即三小时就可以养大一头猪。

3、产品成本低，利润高

在种猪、饲料、设备等条件符合要求的工厂化猪场，猪群生长肥育迅速，一般从出生至出售不超过 6 个月，因此出栏周转快，增重耗料少，房舍利用率高，节省人工，因此可以实现优质高产，降低生产成本，获得高额的利润。

5.2 山西宏明养殖有限公司简介

山西宏明养殖有限公司是以猪养殖、屠宰加工、熟肉制品、饲料加工和产品销售为一体的省级龙头企业，是山西省农业厅示范养殖基地、国家生猪活体储备基地场。该公司从 1998 年兴建起到现在，已形成 5000 头生产母猪的种猪繁殖场和年屠宰加工生猪 20 万头的屠宰厂。该公司出品的“崇康”牌无公害猪肉、熟肉制品，多次获得国家及省农博会“金奖”。

山西宏明养猪场在布局上大体包括生产区、生产辅助区、生活区。

生产区包括各种猪舍、更衣洗澡消毒室、消毒池、药房、隔离猪舍、兽医室、出猪台、称猪台和粪便处理区。生产区是全场的中心，该区应安排在生活区的下风向。猪舍位于上风向，肥育舍位于下风向。育肥舍设在场的一端，有独立的出猪门和出猪通道，出猪通道和出猪台连接，以便装猪出场。为了防疫卫生，生产区独立、封闭。

生产辅助区包括饲料厂、仓库、办公室、水塔、水泵房、淋浴消毒室、机修车间等。饲料厂与饲料仓库可与生产区相连，以便饲料运输。

生活区包括职工宿舍、饭堂、汽车库、文化娱乐室、运动场。考虑到职工的工作和生活环境不受恶臭污染，生活区设在生产区上风向。

5.3 实施效果

连续、集约、全进全出的工厂化养猪生产模式，使得山西宏明养殖场在生产管理上存在很多难题。

工厂化养猪数据量大，数据结构复杂，而且要求及时、准确、完整地提供统计资料，靠传统人工方法进行数据处理难以完成。因为数据之间有时形成交叉结构，有时形成网状结构，数据处理相当困难。工厂化养猪产生的数据资料要求永久保留、安全存放，以备查找。这一点是手工方式很难做到的。比如每周要配种 25 头母猪，每周要有 20~30 头母猪分娩，新生仔猪 200 头左右，相应各生产环节将有 200 头左右的猪只进行转进、转出，这样每周将产生大量数据。仅生产一项一年就将产生数十兆字节的数据。如此大量的数据，靠传统手工方式很难满足对数据处理的准确性、及时性、完整性的要求，难以实现对猪场的宏观管理。

养猪生产中需要进行生产监测，如提示何时进行转舍、配种、分娩等等，往往只有饲养员或配种员才知道每头猪的状况，靠人工难免会出现延误或出错的情况。

在猪场生产报表和每周的例会是个场必做的工作，其目的是领导者要了解场内情况，便于发现问题采取对策。由于上层领导掌握的信息多数来源于基层的汇报，这样必然存在反映情况不及时、不全甚至失真。

利用本系统，山西宏明养殖场在数据、档案、资料管理方面进行加工和分析。由生产线提供每头猪生产情况的原始资料，通过计算机对各猪舍分别进行统计、分析，及时了解到当前各猪舍的存栏信息。库存管理也实现数字化，按入库和出库物品的编号、数量、种类、日期、领货人、价格等资料输入计算机，然后由系统将资料分类累计、列表。这样可以利用计算机的高速度、高准确性进行查询，了解仓库目前各物品的储存量、商品价格及单位成本，从而有利于计划使用及做好购买计划。利用本系统进行监测生产，只要将每头猪最初的状况录入，以后随着时间的变化，系统能进行生产提示，如哪些猪应配种，哪些猪应防疫，哪些猪要产仔等等情况。这样，让更多人都能知道最近要做的重要工作，大家共同关心相互监督。

饲料管理、各种统计报告也应用本系统进行规范化管理和保存资料，保证数据资料的完整性。通过图表的方式形象反映生产，领导者可以从图形报表中获取第一手的资料，全面快速掌控生产情况，如猪只销售情况、全场盈亏情况、猪只健康情况、患病史及趋势等诸多信息，从而做出科学的决策，并极大的降低了人工操作出错的可能性。本系统对提高生产技术水平、管理水平、节约开支、降低成本，提高经济效益等都起着积极的作用。

第六章 总结与讨论

商品猪精细养殖生产管理数字化平台得到了国家“863”计划“数字农业技术与示范”重大专项第六子专题“商品猪精细养殖数字化技术平台”(2003AA209050-6)的支持。系统的实现表明,综合运用数字化技术对商品猪的生产过程进行全方位的管理是可行的、有效的。从系统开发过程来看,该平台有以下创新:

1、系统采用面向对象的分析和设计方法,以商品猪对象为中心,依据实际生产建立问题域模型,对生产情况进行流程模拟和行为模拟,更贴近、直接的描述商品猪的生产过程,便于构造出模块化、可重用的系统。

2、系统采用开放性好的 B/S 体系结构,在伸缩性、可扩展性以及安全性方面具有更大的优势,所有应用程序逻辑在服务器上运行,用户只要获得有效权限,便可以在网络中的不同位置 and 不同客户端上随时申请对信息的操作,方便管理和更新,极大的降低了客户端的负荷符合我国农村实际应用情况。

3、系统采用 MVC 设计模式,将页面表示层、控制层和业务层分开,表示层运用 JSP、CrystalReport 设计,控制层由 Servlet 技术完成,业务层由 JavaBean 进行封装。这使得应用程序的结构更加清晰,通过将代码按照层次划分为用户界面、应用流程控制业务逻辑/数据三个层次,层次间关系明确,增强代码稳定性,有利于三个部分并行开发、加快进度。

4、系统采用 JavaBean 的组件技术,具有可重用、与平台无关、可移植、兼容性、可扩展、创建和使用方便的特点。通过对系统进行分析,在实际的操作过程中,系统需要完成大量的重复性的动作,比如数据库的操作、分页显示等,会在各个不同的模块重复利用。采用面向对象和模块的设计思想,将各个模块抽象为类,充分体现了封装,继承和多态性的面向对象的思想。在 WEB 应用中,通过 JavaBean 与 JSP,Servlet 的结合,可以极大的扩展程序的功能,组件的重用提高了开发的效率和质量,维护也更加方便。

另外,系统采用经济易用高效的 Web 数据库 SQLServer2000,降低了开发的费用,节省了投资,普及性强。加以安全性方面的一些设计,增强了系统的健壮性。

尽管系统在设计和实施上做了以上的考虑,但还需在以下几个方面加以改进:

1、整个系统力求真正做到不懂任何 IT 专业知识的人士,能够完成所有的工作,目前还达不到这种程度,必须先由一定的专业人士利用此系统搭建初步的网站,但在维护和发布上做到了真正的简单易用。

2、由于完成本系统需要大量的专业畜牧知识做支持,在短期内无法做到全而精。目前系统在这方面还有大量的工作要做,随着专业知识的不断积累和在实践中检验,系统会逐渐完善,贴近生产实际。

3、进一步提高系统的缓存机制,由于一个动态页面的速度往往会比静态页面慢 2~10 倍,因此对于一个访问量逐步发展的网站来说,访问速度很快成为一个瓶颈。除了优化内容发布系统的应用本身外,如果能把更新频率比较低的动态页面转存成静态网页来发布,

速度上的提升效果将是显著的，而静态网页如果能被缓存在内存里，访问速度更会比原有动态网页有 2~3 个数量级的提升。

4、在大量用户点击的情况下，会出现服务器当机的情况，这是一个很值得注意的问题，除了提高程序本身的性能，优化数据库等之外，Web 服务器可能也是一个关键的问题所在，现在解决的办法采取定期更新下载、定时重启服务器，可以在一定程度上解决当机问题。

参考文献

1. 柴跃廷等.应用软件系统开发[M].北京:清华大学出版社,1999
2. 陈川等.基于 MVC 设计模式构筑 JSP/Servlet+EJB 的 Web 应用[J].计算机工程,2001,27(11):71~73
3. 陈华军.J2EE 构建企业级应用解决方案[M].北京:人民邮电出版社,2002,282~291
4. 冯辛安.CAD/CAM 技术概论[M].北京:机械工业出版社,1995
5. 冯玉琳等.对象技术导论[M].北京:科学出版社,1998
6. 甘仞初.信息系统开发[M].北京:经济科学出版社,2000
7. 郭丽君.CAD/CAM 技术在企业中的应用[J].潍坊学院学报,2004,(02):103~105
8. 何成万等.MVC 模型及软件框架 Struts 的研究[J].计算机工程,2002,28(6):274~276
9. 梅宏.软件构件和软件构造技术[J].计算机世界,1999,(3):42~45
10. 胡成波.欧盟安全畜产品生产的几点经验[J].中国动物保健,2004,(5):47~48
11. 胡肆农等.养鸡场自动监视监控系统的研制与示范[J].农村实用工程技术,2002,(9):21~22
12. 黄理等.用 JSP 轻松开发 Web 网站[M].北京:希望电子出版社,2001:12~14
13. 姜鸿飞等.面向对象开发方法的最新进展[J].计算机科学,1998,(2):154~161
14. 科技部 863 计划联合办公室.“数字农业技术与示范”重大专项第一批课题申请指南.
http://www.863.org.cn/863_105/applyguide/guide_bio%26agri/200406020015.html(2003-08-20)
15. 孔晓昀.企业 Intranet 的建设[J].水利电力机械,2001,(05):10~12
16. 陆昌华等.20 万只羽蛋鸡规模化生产管理网络信息系统的设计[J].农业工程学报,1998,14(4):182~185
17. 陆昌华.计算机在 20 万只羽蛋鸡场生产管理中的应用研究[J].农业工程学报,1997,13(1):157~159
18. 陆昌华等.利用神经网络改进鸡疾病临床诊断专家系统[J].江苏农业学报,1999,(1):42~46
19. 梅方权.当代农业信息科学技术的发展与中国的对策[J].计算机与农业,2003(1):4~7
20. 宁汝新等.CAD/CAM 技术[M].北京:机械工业出版社,2003
21. 李保明.我国畜牧工程技术发展的现状与趋势[J].中国家禽,2002,(14):5~8
22. 李光志等.对象分析与设计方法比较[M].北京:电子工业出版社,1996
23. 李明.基于 Java 的应用服务器的设计与实现[J].计算机研究与发展,2001,38(5):939~942
24. 李琪.电子商务纵横谈[J].网络与信息,1997,(12):4~8
25. 刘少伯.农业生产与消费战略的讨论.<http://www.agrionline.net.cn>(2003-8-21)
26. 刘少伯.信息时代与畜牧业信息化[J].中国动物保健,2001,(5):25
27. 刘世洪.农场企业信息化研究的特点与前景[J].计算机与农业,1999,(2):9~12
28. 刘世洪.数字农业与农业企业经营管理信息化技术研究[J].计算机与农业,2003,(3):3~5
29. 刘晓华.J2EE 企业级应用开发[M].北京:电子工业出版社,2003
30. 刘勇等.以 CIMS 工程推动企业信息化建设[J].中国科技产业,2004,(1):53
31. 柳松青.面向对象在数据库设计中的应用[J].计算机工程与设计,2002,23(8):51~54
32. 孟庆翔等译.奶牛营养需要(第七次修订版) [M].北京:中国农业大学出版社,2004
33. 农业部畜禽产品质量监督检验测试中心.国内外畜产品质量安全对比分析[J].中国牧业通讯,2003,(13):16~19
34. 乔娟.中国肉类产品国际竞争力研究[M].北京:中国农业出版社,2002,264~278

35. 谯仕彦等译.猪营养需要(第十次修订版) [M].北京:中国农业大学出版社,1998
36. 秦富等.欧美食品安全体系研究[M].北京:中国农业出版社,2003,19~188
37. 全国畜牧兽医总站种畜禽管理处.全国种猪遗传评估方案(试用).2003-3-8.[http://www.chinaswine.org.cn/Display_Content.asp?ID=1499\(2005-3-11\)](http://www.chinaswine.org.cn/Display_Content.asp?ID=1499(2005-3-11))
38. 邵建成.电子计算机在畜牧业中的应用[J].黄牛杂志,1997,23(3):56~60
39. 邵维忠等.统一建模语言 UML 述评[J].计算机研究与发展,1999,(4):385~394
40. 时增善.计算机在猪场的应用[J].中国工厂化养猪二十年论文集,353~357
41. 宋宪一.我国 CAD/CAM 技术应用现状及其发展趋势[J].CAD/CAM 与制造业信息化,2004,(10):9~11
42. 苏建华等.现行奶牛编号在计算机应用上存在的问题[J].中国奶牛,2000,(6):36~37
43. 孙士忠等.浅谈农业企业信息库的建设[J].农业经济管理,1997,(1):51~52
44. 唐桂华.我国 MIS 的开发现状及发展趋势[J].电子科技大学学报(社科版),2002,(02):22~24
45. 唐鸿儒等.企业信息化技术研究[J].计算机应用研究,2003,(11):52~56
46. 王立方等.工厂化猪肉安全生产溯源数字系统的设计[J].江苏农业学报,2004,(4):259~263
47. 王丽群等.基于组件的分布式应用系统的实现[J].吉林大学学报,2002,20(4):77~79
48. 王丽伟等.谈 CIMS 的发展及其应用[J].中国仪器仪表,2004,(10):1~3
49. 吴其庆.J2EE 编程思想与实践[M].北京:冶金工业出版社,2003
50. 徐文胜等.大型数据库性能调整的研究[J].计算机工程,1999,11(5):77~80
51. 薛华成.管理信息系统[M].北京:清华大学出版社,1999
52. 应若平.农业企业信息的发展优势和建设策略分析[J].求索,2003,(2):40~42
53. 于康震等.入世与畜产品质量安全研究(一) [J].中国动物检疫,2003,20(3):1~2
54. 于康震等.入世与畜产品质量安全研究(二) [J].中国动物检疫,2003,20(4):1~4
55. 张延中.浅谈中小企业 Intranet 建设[J].郑州工业高等专科学校学报,2001,(3):45~46
56. 周玉清等.ERP 原理与应用[M].北京:机械工业出版社,2002
57. 赵东等.基于 WEB 的三层结构在智能供应链中的应用[J].计算机工程,2002,28(1):222~225
58. 赵敏.基于 SQL Server 性能调整和测评方法[J].计算机工程,2000,(5):30
59. 赵昕红.欧洲安全畜产品生产控制措施[J].中国饲料,2001,(2):33~35
60. 仲辉.面向对象与结构的系统分析与设计方法应用研究[J].计算机工程与应用,2003,(23):105~116
61. 周卫东,周劲.基于组件的新型分布式客户/服务器系统设计[J].山东大学学报,2002,32(4):308~312
62. 周之英.现代软件工程[M].北京:科学出版社,1999
63. 朱二连等.谈 MRP-II 及其应用[J].地质技术经济管理,1996,(5):53~56
64. 朱明.农业生物环境与设施工程的发展与对策[J].农机科技推广,2004,(5):10~12
65. 朱能武等.计算机技术在畜牧业中的应用[J].计算机与农业,2000,(12):3~5
66. 朱能武等.我国工厂化养猪环境工程技术研究应用现状及发展前景[J].湖北农业科学,2000,(6):56~58
67. Agriculture of National Research Council.Nutrient Requirements of Swine (Tenth Revised Edition) [M].Washington,D.C.USA :The National Academy Press,1998
68. A.L.Azevedo, J.P.Sousa. A Component-based approach to support order planning in a distributed manufacturing enterprise[J].Journal of Materials Processing Technology, 2000,(107):431~438
69. Asbury S,Weiner S R.Developing Java Enterprise Application.New York:John Wiley&Sons, Inc.,2001,223~225

70. Cay Horstmann.Object-Oriented Design & Patterns[M].John Wiley & Sons,Inc., 2004
71. Cheng yongzheng, The Development of the DSS Based on GIS for Regional Agricultural Management. Asian Agricultural Information Technology & Management[C]. Beijing: China Agricultural Sciencetech Press:327~329
72. Deepak Alur, Dan Malks, John Crupi.Foreword by Grady Beech, Foreword by Martin Fowler, Core J2EE Patterns (Core Design Series):Best Practices and Design Strategies, Pearson, 2003
73. ED Roman.Mastering Enterprise JavaBean[M].New York:John Wiley & Sons, Inc.,1999,202~204
74. Grant Larsen.Component-based Enterprise Frameworks[J].Communications of the ACM, 2000,43(10):57-60
75. James Rumbaugh,Ivar Jacobson,Grady Booch.The Unified Modeling Language Reference Manual[M].OMG, 1997
76. J2EE 白皮书,<http://java.sun.com/j2ee/white.htm>
77. Java2 Enterprise Edition Specification, v1.3,final draft,Sun Microsystems,Inc.,2000
78. John Goodson.Java 2 Platform, Enterprise Edition (J2EE) Data Access:The Java DataBase Connectivity (JDBC) Specification and the J2EE Connector Architecture[M].Sun Corp,2002
79. Ma Guilian,Jiang Hongtao and Zhang qin, Agriculture & Precision Agriculture. Asian Agricultural Information Technology & Management[C].Beijing:China Agricultural Sciencetech Press:500~504
80. Mary Kirtland.The Programmable Web Services Provide Building Blocks for the Microsoft .NET Framework[M].The Microsoft Journal for Developers,2001
81. Microsoft SQLServer Performance Tuning and Optimization Strategies,<http://www.sql-server-performance.com/articles-performance.asp>
82. Patrick O' Neil,Elizabeth O' Neil.Database-Principles,Programming, and Performance Second Edition [M].Morgan Kaufmann Publisher,2002
83. PigMap 产品介绍. [http://www.ya-wei.com/diannao.htm\(2005-3-12\)](http://www.ya-wei.com/diannao.htm(2005-3-12))
84. RalphEJohnson.Frameworks=(Components+Paterns) [J].Communications of the ACM, 1997,40 (10):73~78
85. Best Practices For Software Development Teams[M].Rational Unified Process:Rational Software Corp,White Paper,1999
86. R.Neckes,R.Fikes.Enabling Technology For Database Bottlenecks[J].AI Managize,1996,12(3):36-56
87. Robert B Hellman,Irene Grief.Database Performance Optimization and Capacity Planning[J].Business Communications Review,June 1997,28~32
88. Ronald J Norman.Object-Oriented System Analysis and Design[M].Prentice Hall,Inc.,1996
89. Shurson J.Information management for pig profession[J].pig news and information, 1997,18(2):260~275
90. Subrahmanyoun,Allamaraju.J2EE 服务器端高级编程 [M]. 北京:机械工业出版社,2001,(9):67~72
91. Xia Tongshui,zhao Yuanfeng, Developments and Applications of Agricultural Information Technology of China.Asian Agricultural Information Technology & Management[C]. Beijing: China Agricultural Sciencetech Press:146~151

致 谢

在这篇论文完成之际，首先向我的导师刘世洪研究员三年来对我的培养表示衷心的感谢。本文的研究工作，从始至终都凝结了刘老师的辛勤汗水和心血。在我攻读硕士学位三年的时间里，刘老师不仅给予我学术上的指导，在生活上也给予了我很多的帮助。他渊博的知识、丰富的社会经验、宽广的胸怀都让我受益匪浅，从而使我在学业和为人处世这两方面都有了不小的进步。

感谢农业信息研究所和研究生院的各位领导、老师和同学们，他们在我三年学习、生活中给予的帮助和支持。

感谢中国农业科学院农业信息研究所农业信息技术重点实验室的全体老师和同学，特别是钱平研究员、崔运鹏、郑火国，他们对我论文的完成给予了大力的帮助。感谢实验室的苏晓鹭、胡海燕老师对我日常工作学习给予了极大的关怀。感谢王路敬老师、周义桃老师平时对我的教诲，令我在成长路上受益匪浅。

感谢中国农业科学院畜牧研究所熊本海博士、罗清尧，他们严谨治学、认真执着的科研精神永远是我学习的榜样。他们给予我专业上的指导，提供了许多有益的 and 关键性的宝贵指导意见。在此表示衷心的感谢！

感谢于向洪、付彤、赵华、潘月红、朱亮等同学，以及我的师弟郦晶、朱海鹏，他们给予我无私的帮助并和我一起度过快乐、充实的学习时光！

我还要特别感谢我的父母、女友及关怀我的亲朋好友。感谢他们在我三年学习过程中给予的理解与支持。没有他们在物质和精神上的大力支持，我将无法顺利完成学业。

谨以此文献给我人生路途中培养、指导我的各位老师，献给我的家人、关爱我的朋友以及帮助我的各位同仁。

本研究得到国家“863”项目资助，在此谨致谢意。

作者简历

苏希, 男, 1979 年 2 月生于山西省太原市。2001 年 7 月毕业于沈阳工业学院计算机应用专业, 获工学学士学位。2002 年 9 月考取中国农业科学院研究生院管理科学与工程专业硕士研究生。

硕士期间发表的论文:

1. 苏希. 农业信息技术的发展与应用前景[J]. 计算机与农业, 2003(10): 6~9
2. 苏希, 刘世洪. 运用 J2EE 框架的商品畜禽精细养殖技术平台设计[J]. 农业网络信息, 2004(12): 15~17
3. 苏希, 刘世洪. 农业企业管理信息系统构建方案的思考[J]. 农业图书情报学刊, 2005, 17(2): 94~196