

分类号: TP311

单位代码: 10422

密 级:

学 号: 200812656



山东大学

硕士学位论文

Shandong University Master's Thesis

论文题目: 灾害天气综合数据库系统设计与实现

The Design and Implementation of the

Integrated Database System of Severe Weather

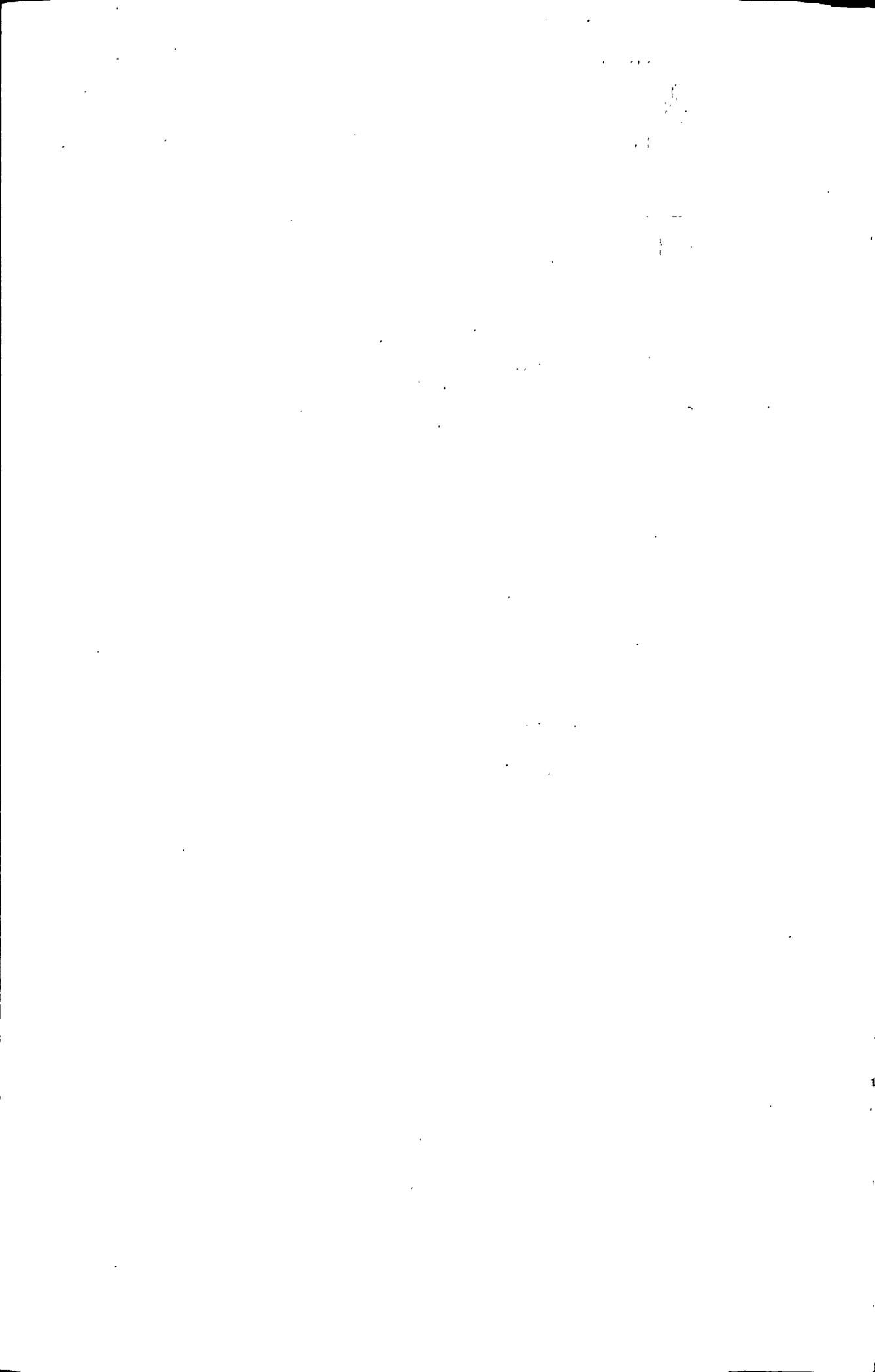
作 者 _____ 刘 岩 _____

专 业 _____ 软件工程 _____

导 师 _____ 张彩明 教授 _____

合作导师 _____

2010 年 4 月 10 日





原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

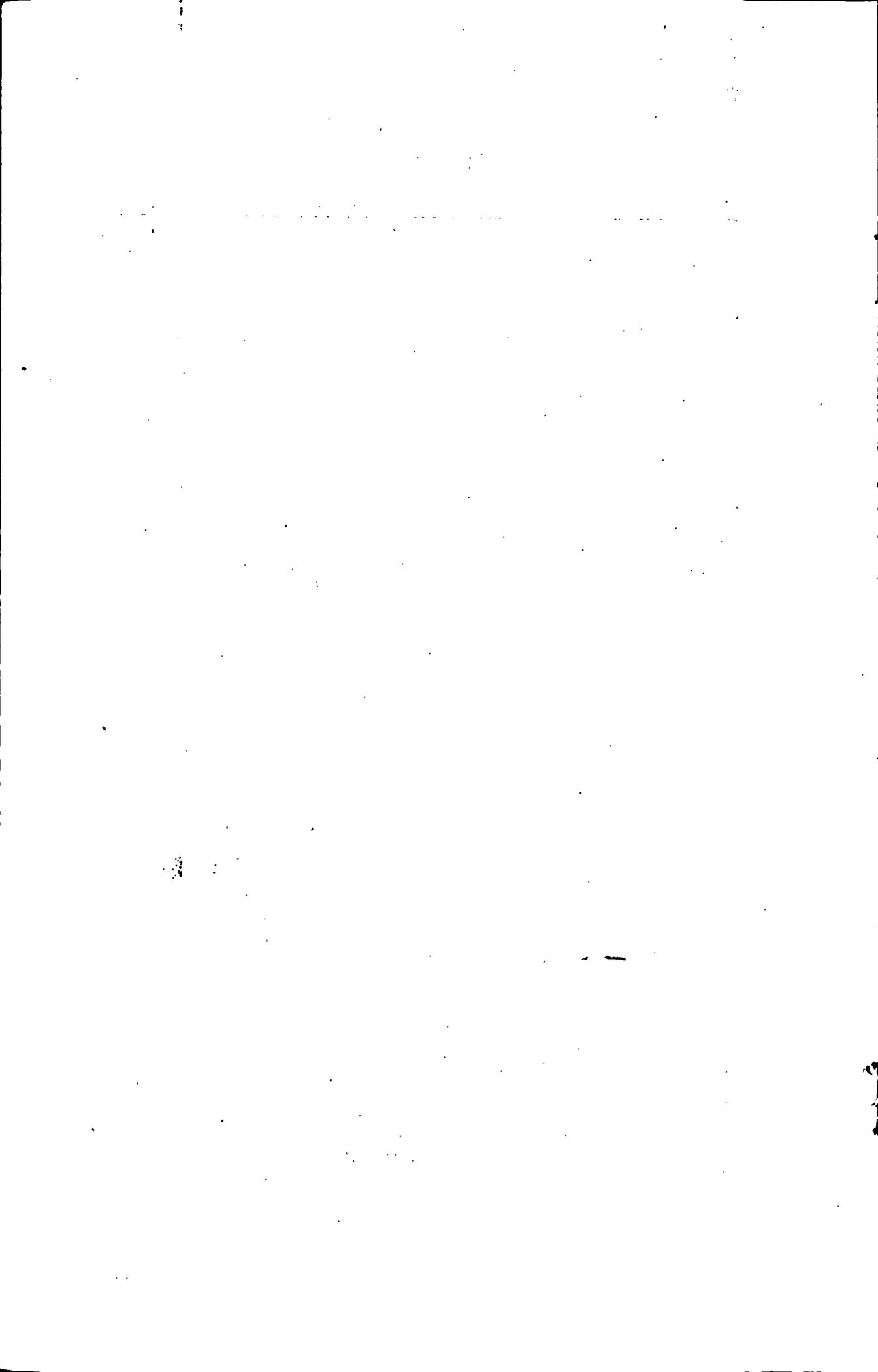
论文作者签名： 刘岩 日期： 2010.4.10

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名： 刘岩 导师签名： 柳学明 日期： 2010.4.10



目 录

目 录.....	i
摘 要.....	I
第 1 章 绪论.....	1
1.1 灾害天气综合数据库系统开发背景.....	1
1.2 国内外研究现状及发展趋势.....	3
1.3 解决的主要问题.....	5
1.4 本文的主要工作.....	5
1.5 论文的组织结构.....	6
第 2 章 灾害天气综合数据库系统需求分析.....	8
2.1 灾害天气综合数据库系统概述.....	8
2.1.1 系统目标.....	8
2.1.2 灾害天气综合数据库系统解决方案及系统逻辑.....	8
2.2 灾害天气综合数据库系统需求获取模式.....	9
2.3 数据分析模板.....	11
2.4 灾害天气综合数据库系统功能性需求.....	13
2.4.1 系统的参与者和目标.....	15
2.4.2 业务流程及用例.....	16
2.5 灾害天气综合数据库系统非功能性要求.....	23
2.5.1 系统运行时效性.....	23
2.5.2 可靠性要求.....	23
2.5.3 维护性要求.....	23
第 3 章 灾害天气综合数据库系统架构设计.....	25
3.1 系统设计目标和原则.....	25
3.2 灾害天气综合数据库系统技术架构设计.....	26
3.2.1 系统总体功能划分.....	26
3.2.2 系统逻辑架构.....	28

3.2.3 系统功能流程.....	29
第4章 灾害天气综合数据库系统详细设计	31
4.1 灾害天气综合数据库系统存储规划设计	31
4.1.1 数据文件存储区域设计.....	32
4.1.2 数据库逻辑模型设计.....	34
4.2 系统建模	40
4.2.1 系统静态建模	40
4.2.2 系统动态建模	47
第5章 灾害天气综合数据库系统实现	49
5.1 系统的总体界面与配置	49
5.2 数据预处理功能的实现	51
5.2.1 数据预处理功能信息流程.....	51
5.2.2 数据预处理功能实现.....	51
5.3 数据存储管理功能实现	52
5.3.1 数据获取子功能的实现.....	52
5.3.2 元数据提取子功能的实现.....	54
5.3.3 数据解析入库子功能实现.....	55
第6章 结论.....	64
参考文献.....	65
致 谢	67

CONTENTS

CONTENTS	i
ABSTRACT	I
Chapter 1 Introduction	1
1.1 Development background of the Integrated Database System of Severe Weather	1
1.2 Latest state of technology and trend in development	3
1.3 The main problems need to be resolved in this paper	5
1.4 The main work of this paper	5
1.5 The structure of this paper	6
Chapter 2 The requirement analysis of the Integrated Database System of Severe Weather	8
2.1 Introduction to the Integrated Database System of Severe Weather	8
2.1.1 Goal of the system	8
2.1.2 Solution and system logic.....	8
2.2 Requirements capture mode	9
2.3 Data Analysis Template	11
2.4 Functional requirement.....	13
2.4.1 Acters and their goals	15
2.4.2 Business process and the use cases	16
2.5 Non-functional requirement	23
2.5.1 The timeliness of system operation	23
2.5.2 Reliability requirements	23
2.5.3 Maintenance requirements.....	23
Chapter 3 Construction design of the Integrated Database System of Severe Weather	25
3.1 Designing goal and principle of this system.....	25
3.2 Technology construction design	26
3.2.1 General function division of this system	26
3.2.2 Logical construction of this system.....	28
3.2.3 System procedures	29
Chapter 4 Detail design of the Integrated Database System of Severe Weather	31

4.1 Design of storage planning	31
4.1.1 Design of the data file storage area	32
4.1.2 Design of database logical model	34
4.2 System modeling	40
4.2.1 Static modeling	40
4.2.2 Dynamic modeling	47
Chapter5 Implementation of the Integrated Database System of Severe Weather	49
5.1 System interface and configuration	49
5.2 Implementation of data pre-processing	51
5.2.1 Procedures of data pre-processing	51
5.2.2 Implementation	51
5.3 Implementation of the storage management function	52
5.3.1 Implementation of the data acquisition	52
5.3.2 Implementation of meta-data extraction	54
5.3.3 Implementation of data storage	55
Chapter6 Conclusion	64
References	65
Acknowledge	67

摘 要

我国地处亚洲东部,地形分布特征极其复杂,灾害天气频发,每年给国民经济人民生活带来巨大损失^[1]。灾害天气综合数据库系统正是中国气象科学研究院围绕国家需求,为研究引发暴雨、大风、冰雹以及沙尘暴等灾害天气的形成原因,提升灾害天气监测与预测水平而对卫星、雷达、常规资料、数值预报、973项目数据等众多海量气象观测数据进行科学合理的存储管理,并为气科院各级科研人员和公众提供方便快捷的数据在线服务而投资建设的综合性数据库系统。

灾害天气综合数据库系统的核心在于灾害气象信息数据存储。种类多、范围广、实时性强、海量数据、类型复杂、科学型是气象数据的特征,如何进行有效地存储规划与选择合理的管理模式,一直是气象数据存储领域的重大课题。为此,本文首先简要描述了灾害天气综合数据库系统的开发背景、国内外气象数据资料存储服务领域的现状及发展趋势,阐述解决的主要问题和完成的主要工作。在此基础上,借鉴国外类似系统的开发经验,采用统一建模语言为分析设计的描述语言,对灾害天气综合数据库系统进行了需求分析和系统设计,以数据存储为主线详细介绍了灾害天气气象数据存储规划策略,分析设计并实现了系统的自动化存储管理等应用。

本文着重介绍灾害天气综合数据库系统的构建及在其存储服务实现过程中所采用的关键技术和算法,主要包括:

第一,在充分调研和深入理解气象科学数据标准的基础上,获得用户需求和数据处理业务流程,包括功能性需求和非功能性需求,并通过分析总结出合理存储规划的设计和开发思路。

第二,对现有的历史气象数据进行整理和加工,实现全部元数据和部分要素数据入库工作。

第三,按照功能将系统分解为七大模块,结合工作重点对部分模块进行详细的分析与设计。在系统分析设计过程中,对需求获取的方法、数据特性卡分析手段以及各阶段UML工具的使用等进行了规范。

第四,关注对系统性能有严格要求的检索服务,通过数据检索对比试验,完

成了对后台数据库性能的测试评估，并以此为据着重进行了存储规划设计。

第五，设计构建数据表对象模型，给出系统配置信息，以常规探空数据为例，实现相关产品数据入库的完整处理逻辑。

最后，对灾害天气综合数据库系统的设计和实现进行了总结。

关键字：灾害天气；数据库；海量数据；存储规划；存储管理

ABSTRACT

China is located in eastern Asia. The extremely complex topographic distribution characteristics and the weather disasters may take place every year which cost a great loss to the national economy and people's lives. Integrated database system of severe weather is designed by Chinese Academy of Meteorological Sciences(CAMS), according to the nation's needs and in order to research the cause of torrential rains, strong winds, hail, dust storms and other weather disasters, and to improve the level of monitoring and prediction of weather disasters. For the reasons given above, CAMS invests the construction of this integrated database system which can provide scientific and reasonable storage management of satellites data, radars data, general observation data, numerical forecasting, 973 project data, and many other mass meteorological observation data. After completion, the system will provide convenient online services to both the CAMS researchers at all levels and the general public.

The core of the integrated database system of severe weather is the storage management of meteorological data. The feature of meteorological data is variety, wide range, strongly real-time, massive data, complex types and scientific. How to effectively choose the right storage planning and management model has been a major issue of the domain of meteorological data storage. Therefore, this paper describes the development background of the integrated database system of severe weather, the situation and trend among the fields of meteorological data storage services, and the depiction of the main problem to be finished and completed. After that, foreign experiences in the development of similar systems are referenced in this article. Which using the Unified Modeling Language for analysis and design of the integrated database system of severe weather, needs analysis and system design using data storage as a main line of details of the severe weather meteorological data's storage planning strategy, analysis, designs and implementations of the automatic storage management applications.

This article focuses on the key techniques and algorithms used in the process of the implementation in the construction of the integrated database system of severe weather, which including:

First, on the basis of full investigation and in-depth understanding of

meteorological science data standards, users' needs and data processes, which including functional requirements and non-functional requirements, and then a reasonable conclusion of storage planning design and ideas of development was drawn by analyzing.

Second, processing has done with the existing historical meteorological data, to storage all of the meta-data and some elements of data.

Third, this system is divided to seven main modules, part of which are analysis and designed in detail according to the focus of the task. The demand access methods, the data analysis tools as well as the cards of data characteristics and all stages of the use of UML tools are standardized in the systematic analysis of the design process.

Fourth, data retrieval performance has been focused on as one of the most important requirements for access services. By using the comparison test, the back-end database performance testing and evaluation have been completed. And the result of the test was used for the design of the storage planning.

Fifth, data sheet object models have been designed. In the implementation part of this paper, configuration information of the system have been given, processing logic of data storage will be achieved with an example of the complete procedure of storage of the conventional air sounding data.

Finally, design and implementation of the integrated database system of severe weather are summarized.

Keyword: Severe weather; Database; Massive data; Storage planning; Storage management

第 1 章 绪论

1.1 灾害天气综合数据库系统开发背景

中国气象科学研究院 (CAMS, 即 Chinese Academy of Meteorological Sciences, 以下简称气科院) 隶属于中国气象局, 其前身为 1956 年 8 月成立的中央气象科学研究所, 1978 年更名为中央气象局气象科学研究院, 1991 年更为现名, 2001 年起进入科技部支持的国家公益类研究院。气科院是我国大气科学领域学科种类最多、规模最大的科研机构, 是以研究雷电防护与大气探测、人工影响天气、灾害天气、气候与气候系统、生态环境与农业气象、数值模式以及大气成分等为主攻方向的我国大气科学综合研究基地, 主持和承担了多项国家科技攻关项目、国家自然科学基金重点和重大项目、国家重点基础研究发展规划项目 (973 项目)、国家重大专项以及部委重大研究项目等, 研究内容涉及到大气科学及相关学科的各个方面^[2]。

灾害天气国家重点实验室 (LaSW, 即 State Key Laboratory of Severe Weather) 是在原中国气象科学研究院强风暴实验室和数值预报中心基础上组建的, 2005 年 3 月, 实验室正式进入建设阶段, 2007 年 8 月实验室通过了建设期的验收, 2008 年 3 月正式进入国家重点实验室的行列。

我国地处亚洲东部, 地形分布特征极其复杂, 极端天气频繁, 为世界上遭受自然灾害最多的国家之一。而灾害天气, 主要包括暴雨、台风、冰雹、沙尘暴等, 每年都给国民经济和人民生活财产带来巨大损失。近年来, 在全球变暖背景下, 我国气象灾害和极端天气事件更加频繁发生, 造成的社会经济损失日趋增加, 成为影响国民经济发展、社会进步的重要不利因素。因此, 提高灾害天气的监测与预测水平不仅是各级政府部门指挥防灾减灾最为迫切的需要, 同时也对增强我国防灾减灾的总体能力, 确保我国社会与国民经济可持续发展具有极为重要的意义, 是国家的重大需求。

2005 年以来组建的灾害天气国家重点实验室正是立足国际前沿, 围绕国家需求, 定位于应用基础研究, 重点研究领域为引发暴雨、大风、冰雹等气象灾害的中尺度强对流系统以及沙尘暴等灾害天气, 建成我国在灾害天气监测与预测领域

的研究基地,充分利用技术先进、设施完备的工作平台,集中力量分析积累起来的大量的气象观测数据和资料,进一步揭示灾害性天气发生发展的规律和形成机理,提高对灾害天气的有效监测能力和定量预报能力。

由于体制等方面的原因,目前气科院灾害天气数据的存储管理仍沿用组建之初所使用的老系统,由人工进行处理,采取单一的文件式存储,缺乏统一的管理与调控,数据共享程度较低,数据查询效率较差。天气资料尤其是典型灾害天气个例资料的存档与共享缺乏标准与规范,相当一部分数据存储于档案柜。即便已建成的部分数据库也缺乏可供遵循的标准,数据分类存储、加工能力较差,并且没有引入元数据的概念,这样无论在数据共享还是数据维护方面都带来了极大的困难。另外,由于各式气象数据的存储流程被割裂为几段,需要人工衔接。系统在设计过程中没有遵循用户的业务操作习惯,导致数据存储与服务效率极为低下。以上问题产生的原因,主要是缺少数据存储整体规划、在系统实施过程中对业务及功能需求的把握不准确、系统的设计缺乏整体考虑和灵活性等。

灾害天气国家重点实验室研究的数据范围主要囊括了卫星、雷达、常规资料、数值预报等众多种类的观测数据,并承接如南方暴雨监测预测研究数据等 973 项目数据,这些数据的数据量非常巨大,因此对其进行科学合理的管理,并为科学研究人员提供方便高效的服务就显得至关重要。在这种背景下,灾害天气综合数据库系统建设项目应运而生。

IDSSW (Integrated Database System of Severe Weather 灾害天气综合数据库系统)负责制定常规资料、数值预报、雷达、卫星等基础数据以及灾害天气实验室各项目数据的存储管理策略,根据策略对各类数据进行统一存储与管理,并以此为基础,为气科院各级用户提供基础数据和项目数据的在线查询和下载服务,同时也为外部用户提供一定的查询和下载服务。

新系统建成后,在存储上实现基于 Internet/Intranet 的气象数据资料自动化管理,对中国气象局信息中心推送而来的各种文件数据实现文件存储与部分数据入库存储两种存储方式。本文的重点就是要根据上述要求设计完成对灾害天气数据库的平台开发,研究并应用合理的数据库存储方式,保证应用于气象研究的海量数据有序存储,并为客户提供可靠、快捷、方便的在线服务功能。同时也对未来各省市灾害天气数据中心业务数据存储管理有较强的指导作用,因此具有非常重要的实践意义。

1.2 国内外研究现状及发展趋势

资料显示,气象数据是气象预报和进行气象科学研究的基础。因此,历史数据、实时数据以及相关的气象数据存储与管理自然就成为各类气象应用系统的重要组成部分。在灾害天气数据库系统中,气象数据信息的数据存储与管理框架是系统建设的重要内容。同时,气象数据作为一类科学型数据,其数据存储模式应是标准化、规范化、高可靠、大容量的数据存储与稳定、高效的管理体系,这是系统得以高效运行的关键^[3]。

对于种类多、范围广、实时性强、数据量大、类型复杂的气象数据,应该选择有效的存储与管理模式。目前常用的 Oracle、Sybase 以及 Informix 等数据库系统,主要适用于商业事务数据的存储管理以及检索查询与分析等。由于气象数据类型的复杂性以及巨大的数据量,若采取单一商业数据库,其适应性将受到很大限制^[4]。目前国内外许多科学型数据大都采用文件系统作为存储方式,不同的科学数据采用不同的格式,同时有相应的存储管理体系用以完成数据的读、写以及检索。

在气象数据资料存储服务领域中,国外起步较早,并认为文件系统是比较适宜的存储方式。例如美国的 AWIPS(Advanced Weather Information Processing System, 进阶天气交互处理系统),其格点数据使用 NetCDF 格式存储,而云图、雷达数据则是直接存储了原始的数据文件^[5]。我国气象部门统一使用的 MICAPS(Meteorological Information Comprehensive Analyzed Processing System, 气象信息综合分析处理系统),也采用了类似的存储方式,在数据存储与管理体制中应用与国际标准接轨的设计思想,文件格式多为二进制文件(BIN)与文本文件(TXT)。数据量大是气象数据的显著特点,以文件形式存储的数据可以利用先进的文件压缩技术对其进行压缩,以大大节省磁盘空间。相应的数据管理系统,可以方便地对数据库进行各项操作,而且只需提供特定的特性字段数据即可方便、快捷地检索到所需数据文件^{[6][7]}。

但是,随着气象分析研究技术的日新月异,对数据内容的综合处理已经成为数据存储与科学研究之间日渐扩大的鸿沟,灾害天气的综合分析研究与预报更是如此。就目前而言,气科院乃至中国气象局越来越多的科研项目涉及对不同区域、不同站点、不同时次采样数据的综合分析,尤以本文涉及的灾害天气为代表。因

此单一的文件系统、文件式存储显然已经无法满足科研人员的要求。

在国外有许多成熟的软件和气象服务平台能提供较多的气象服务,应用计算机技术实现气象资料的数字化、表格化、图形化,能很好的将气象资料经过加工处理整合在一起,创立气象数据信息共享平台。在国内一些地方气象部门也已经开始着手气象信息服务平台,气象监测数据存储、气象预报研究、气象信息公共服务的业务系统也很多。然而建立一个灾害天气综合数据库要用到很多技术,能够自主开发的,真正用到实处并能带来很大方便的气象信息综合数据库系统在国内尚未实现^[8]。

国内的气象机构近年已开始构建全面的自动化存储管理,因此气科院借鉴国外类似系统的经验,在2009年提出以文件存储与数据库存储相结合的方式,研发灾害天气综合数据库系统,对气象科学尤其是灾害天气研究各项目所需气象数据资料实现自动化存储管理与存储优化。

灾害天气数据库系统的核心在于灾害气象信息数据存储。一般来讲,灾害天气数据库系统指把应用于研究工作的灾害天气数据库、有关的硬件、软件和人员组合起来提供服务的系统,包含数据库管理系统(DBMS)、支持DBMS的软硬件环境、灾害天气数据库、用户及其应用程序等。

当前,已经有一些气象相关的软件、系统、网站能为气象科研人员提供方便,同时对广大人民群众的生活也提供了很多便利。但是这些服务共享平台系统运行不够稳定,安全性不高,运行速度不够快,尤其在数据存储、数据检索时存在着效率底下的问题。这类系统往往开发周期长、升级难度大、不容易移植,因此推广维护比较困难。在各类气象信息的加工、处理和发布上,也没有一定的逻辑性,缺乏系统的管理与维护。计算机及网络技术的迅猛发展,数据库、动态信息发布、资源高度共享等技术的应用,对气象信息综合数据库平台的整体性能和功能的提升具有十分重要的推动作用。

目前绝大多数大型应用系统采用B/S分层架构,使系统具有更好的可扩展性和可维护性。整个系统可以分为四层结构:客户端—表现层—应用服务层—数据服务层,这四层分别由浏览器—网络服务器—应用服务器—数据库服务器构成。这样的多层软件架构确保了系统的扩展性和适应性。通常选择微软的SQL server或者Oracle为数据库系统。同时,基于这种多层结构,能够很方便得应用智能负载

均衡与集群等技术实现系统服务能力的扩展。

1.3 解决的主要问题

灾害天气综合数据库系统是面向气科院各级科研人员，建立在中国气象局 MICAPS 系统之上的子系统，该系统由国家气象信息中心推送而获得气象信息数据，并对数据进行识别、转存与解析，最终对灾害天气相关的气象科学研究提供可靠的支持。

由于国内气象信息综合数据库系统的发展尚处于一个比较初级的阶段，并没有成熟的系统和规范可以参考，因此如何根据用户需求、类型复杂的气象数据标准设计满足气科院科研人员所需要的灾害天气综合数据库系统，完成数据存储规划与数据存储管理，以提高检索服务效率，是本文要解决的主要问题。

第一，在充分调研和深入理解气象科学数据标准的基础上，获得用户需求和数据处理业务流程，包括功能需求和非功能需求，并通过分析总结出合理存储规划的设计和开发思路。

第二，系统分析、设计过程中的面向对象技术和 UML 技术的使用。同时在设计方面，要充分考虑气象数据类型复杂、数据量巨大等业务特点，设计满足系统的高度适应性、可扩展性和易维护性。在系统整体上着重考虑系统的可靠性和高效性。在系统的数据库设计中要保证数据的准确性与处理流程的可追溯性。

第三、系统核心业务——存储规划与存储服务的实现。在系统设计的基础之上，应用设计好的数据模型实现由数据获取到日志记录一系列业务流程，解决基于多数据源入库请求、并发数据检索请求的效率问题。

1.4 本文的主要工作

本文在现有灾害天气信息系统基础之上，通过分析气科院灾害天气国家重点实验室各项数据存储服务的实际需求和业务流程，结合对气科院各级科研项目数据标准以及中国气象局信息中心数据分类的调研，使用目前最为稳定的 AIX Unix 为系统平台，以 WebSphere 为应用服务器，使用 UML 进行系统分析，Java 作为程序设计语言，应用 J2EE 技术架构，Oracle 9i 为后台数据库，设计和实现灾害天气

综合数据库系统。

本文着重介绍灾害天气综合数据库系统的构建及其存储服务实现过程中所采用的关键技术和算法，主要包括：

第一，对现有的历史气象数据进行整理和加工，通过调研所得各种标准对数据质量进行保证；完成对灾害天气综合数据库系统所涉及的全部历史数据完成数据预处理和文件转存，完成全部元数据和部分要素数据入库。

第二，在系统分析设计过程中，对需求获取的方法、数据特性卡分析手段以及各阶段 UML 工具的使用等进行了规范。

第三，关注对系统性能有重要要求的检索服务，完成了对后台数据库性能的测试评估，并以此为据着重进行了存储规划设计。

第四，构建设计数据表对象模型，实现相关产品数据入库的处理逻辑。

最后，对灾害天气综合数据库系统的应用情况作了简单介绍，并对系统的设计和实现进行了总结。

1.5 论文的组织结构

第1章绪论，主要描述灾害天气综合数据库系统的开发背景、国内外气象数据资料存储服务领域的现状及发展趋势，本文解决的主要问题和完成的工作。

第2章灾害天气综合数据库系统需求分析，主要进行灾害天气综合数据库系统的需求分析。首先进行了灾害天气综合数据库系统目标的概述。并按照功能性需求和非功能性需求两个类别，采用了用例等方法进行了描述。

第3章灾害天气综合数据库系统概要设计，主要进行系统的架构设计。首先对系统的设计目标和原则进行了阐述。其次，在技术架构设计中，分别按照总体功能划分和逻辑架构对系统进行设计，并从系统功能流程的角度给出系统的业务流程设计图。

第4章灾害天气综合数据库系统详细设计，主要进行数据库存储规划和数据存储管理的详细设计。使用包括类图、时序图和协作图等工具完成对所负责系统存储服务的详细描述。

第5章灾害天气综合数据库系统的实现，通过一个完整的数据入库示例，对系

统存储管理模块的算法实现进行了描述。

· 第6章对论文进行了总结，并对系统的进一步提升提出了改进意见。

第 2 章 灾害天气综合数据库系统需求分析

2.1 灾害天气综合数据库系统概述

伴随着信息化的发展与计算机技术的不断革新,气象信息数据库系统在国内外国气象部门已经历多年快速的发展,但本文介绍的灾害天气综合数据库系统在中国气象部门仍处于探索与起步阶段。所以系统的设计与实现必须立足于气象科学研究的实际情况,首要针对目前气科院科研人员的要求,充分调研科学型数据存储的特点,结合中国气象信息中心 MICAPS、MARS 存储管理等系统的经验,为灾害天气数据存储提供科学合理的管理系统,同时面向气科院各级科研人员提供操作简便的系统支持。

2.1.1 系统目标

灾害天气综合数据库系统的建设需要实现以下各项目标:

1. 实现历史气象数据的统一整理与系统录入;
2. 制定合理的存储规划,规范各项业务数据文件分类分级存储;
3. 提高实时气象数据获取转存的自动化,保证海量数据入库的数据一致性与速度;
4. 规范数据入库的流程;
5. 实现数据入库的日志监控,确保入库数据准确无误;
6. 提供便捷的数据服务支持,简化操作复杂度。

2.1.2 灾害天气综合数据库系统解决方案及系统逻辑

灾害天气综合数据库系统负责气象科学信息数据入库管理,整体系统采用 B/S 结构、中间件和 Oracle 数据库的体系结构。前台采用 JSP 生成浏览器页面,后台使用 Java 主要对数据库进行各项操作。采用 MyEclipse 作为开发平台,WebSphere 作为服务器中间件,后台使用的是 Oracle 9i 数据库,整个系统运行在 IBM AIX Unix 5.1 系统下,实现高性能数据服务。

简单的讲,系统由浏览器—web 服务器—应用服务器—数据库服务器组成。出

于对数据安全性和高可靠性的要求，系统建成后将同时部署内外网两套应用到实际的项目之中，内网主要针对气科院内部的科研人员提供数据支持，外网则面向普通用户提供有限数据资源。两套应用中的数据库将采用同步技术提供服务，因而两套设施提供的服务完全一致，同时也加强了系统整体的可扩展性，以方便满足日后系统需求的变更。鉴于内外网系统完全相同，本文仅就内网系统的开发做说明。

气科院为系统提供的两台 IBM 服务器作为项目的主要基础设施，有效地利用两台服务器可大大优化项目的逻辑过程。下图简单说明了这一方案(图 2-1)。

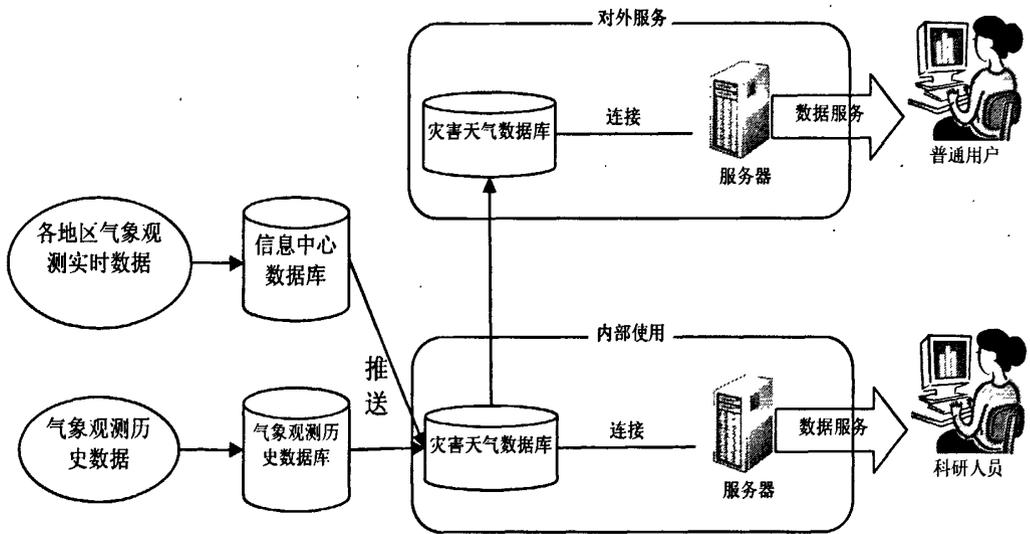


图 2-1 灾害天气综合数据库实施系统方案

2.2 灾害天气综合数据库系统需求获取模式

什么是需求？简单讲，需求就是系统必须提供的能力和必须遵从的条件。根据最新的软件工程标准术语表，需求被定义为：

用户为解决某个问题或达到某个目标而需具备的条件或能力。

系统或系统组件为符合合同、标准、规范或其他正式文档而必须满足的条件或必须具备的能力。

上述第一项或第二项中定义的条件和能力的文档表达^[9]。

上述定义既包含了用户对需求的看法，也代表了开发人员的观点。按照传统软

件工程理论的观点，需求分析的目标就是要确定“做什么？”。因此，寻找、沟通和记录什么是用户真正所需，并能够将这些清楚地讲解给用户和开发团队各成员就成为需求分析最大的挑战^[10]。

由于本系统是面向气科院各级工作人员的专用系统，信息量与各课题组要求较为复杂，为获取正确的需求分析，开发时我们采用了常规需求获取方法、统一过程方法相结合的需求获取模式^[11]：

1. 建立分析小组。在系统开发之初，我们对用户的业务和术语并不熟悉，用户对我们的开发流程也不熟悉，因而用户提供的需求信息，在我们看来只是零散和片面的，没有条理。为此专门建立了由灾害天气国家重点实验室科研人员为主，系统分析员和熟悉气象信息化的领域专家共同组成的联合分析小组进行沟通，对系统需求进行共同探讨。
2. 用户访谈。在进行需求分析时，访谈的对象包括对方高层用户，如各级实验室科室领导，也包括各级科研人员等直接用户。根据用户使用系统的功能、频度、优先级和熟练程度等方面的差异，将他们分为领导用户、管理员用户与一般科研人员等几类，针对每类用户采用了个别座谈和小组会议等不同形式，调研各类用户的习惯特点及任务状况，采用循序渐进、步步逼近的原则，了解他们对新系统功能的期望与要求。
3. 问题的分析与确认。在需求分析的过程当中，我们发现用户所述的需求往往是不断变化的，因此不能期望他们在一两次访谈中就对本系统的需求做出明确的阐述。通过访谈后及时整理、去掉错误与无关的信息，留下有用的内容，在接下来访谈中与用户进行协商、问题确认，并不断将我们需要了解的问题细化。
4. 统一过程方法。确定系统开发过程中各个环节应该做什么、怎么做、何时做以及为什么做，通过描述过程，定义如何使用人力、机器、工具和信息等资源的一些规则来完成某个确定的目标，为用户的问题提供解决方案。这一方法包括了初始、细化、构造和交付四个阶段。其中，在初始阶段通过确定系统中绝大多数角色和用例、划分出主要子系统、分析执行风险、考虑时间经费等因素和制定开发计划来确定项目的可行性。在细化阶段，通过对需求风险、技术技能风险的分析，制定构造阶段计划，对每个用例

进行了细化,分析用例的处理流程、状态细节及可能发生的状态变化。

2.3 数据分析模板

灾害天气数据库系统所要管理的数据不但种类繁多、格式复杂多样,数据量也十分巨大。因此,系统最重要的工作就是实现对这些数据科学合理的管理,这就要求事先对这些数据的数据格式、数据内容、数据到达频率、数据量、存储要求等信息进行细致深入的分析,并在此基础上合理规划数据存储的逻辑和物理结构。

灾害天气综合数据库系统建设项目所管理的数据包括卫星、雷达、探空、常规资料、数值预报等基础资料数据以及不同项目产生的暴雨、沙尘等个例数据,其中每一大类数据又包括若干数据分支;为实现对各类数据都能进行深入、细致、有序的分析,首先特制定了数据调研特性卡模版,以便收集各类数据的数据特性;然后根据这些特性,对所管理数据进行整体分析,并在此基础上进行数据存储规划。其中数据调研特性卡模版样式如下:

表 2-1 数据调研特性卡模板

数据类型			
数据名称			
数据描述	数据内容概述		
命名规则	✓ 命名规则:		
数据获取方式	(主动推送)		
数据提供方	数据提供单位或系统名		
源数据存放目录			
数据到达频率	半个小时	文件个数/天	
每个文件数据量	每个数据文件的大小		
数据保存策略	保存周期、流转情况、到期后处理策略等		
数据的关联关系	与数据相关的描述文件、质量文件等		
文件的保存形态	1) 文件+元数据		
数据的元数据字段			
数据实体结构	描述需解析入库文件的内部结构,内容包括:		
数据实体记录数	需解析入库单个文件的记录数		

数据在线服务要求	1) 查询要求
数据的分类原则	从上至下描述数据分类层次,
数据存放目录规则	比如是否要求按月份、日期等规划数据子目录
特殊数据特性属性说明	1. 对数据特性中特殊的或理解有歧义的属性进行备注

在需求分析操作过程中, 根据以上模版, 通过向气科院各级相关人员收集各类数据的特性信息。下面列表给出了探空类数据进行调研时收集到的数据特性:

数据类型	常规观测资料		
数据名称	探空与加密探空资料		
数据描述	全国探空与加密探空资料, 判定一个数据文件是正常观测数据还是加密观测数据根据文件名中的时间判定, 02、14 为加密, 08、20 为常规		
命名规则	<ul style="list-style-type: none"> ✓ 命名规则: high_yyyymmddhh.txt ✓ 文件名各字段解释: high: 文件名固定部分, 表示探空资料 yyyyymmddhh: 表示资料产生的时间, 格式为“年月日时” 扩展名: 文件扩展名为“txt”, 文本文件 ✓ 文件名举例: high_2009070600.txt 		
数据获取方式	推送		
数据提供方	国家气象信息中心		
源数据存放目录	/data/base/high/yyyymm 每年文件数: $2*365+2*180 \approx 910$ 个		
数据到达频率	12 小时 (或 6 小时)	文件个数/天	常规观测每天 2 个文件, 加密观测每天 4 个文件
每个文件数据量	每个文件约 500KB		
数据保存策略	长期保存。数据要素入库, 提供在线查询。		
数据相关静态文档	文档名称: 文件是否要解析入库? 解析入库		
数据动态关联文件	无		
数据实时汇图函数	数据显示时, 可根据用户要求使用此函数实时汇制图形图像		
文件的保存形态	数据库存储+元数据		
数据的元数据字段	数据人类 (基本数据 BASE)、数据类型 (常规资料)、数据子类 (探空)、数据属性 (加密/常规)、数据产生日期 (yyyyymmdd)、产生时间 (hh)、文件格式 (txt)、文件名、文件存储目录、是否入库成功、缺测或无效 (999999)、数据是否有效、备注		
数据实体结构	<ul style="list-style-type: none"> ✓ 实体结构: 文件头: 年月日 时 探空站数 文件体: 区站号 经度 纬度 海拔高度 层数 第 1 层气压 高度 温度 露点 风向 风速 第 n 层气压 高度 温度 露点 风向 风速 ✓ 字段说明: ◆ 实测字段说明 区站号: 整数 经度: 度, 浮点, 小数点后四位 纬度: 度, 浮点, 小数点后四位 海拔高度: 米, 浮点, 小数点后一位 第 n 层气压: 百帕, 浮点, 小数点后一位 		

	<p>高度: 米, 浮点, 小数点后一位 温度: 度, 浮点, 小数点后一位 露点: 度, 浮点, 小数点后一位 风向: 度, 浮点, 小数点后一位 风速: 米/秒, 浮点, 小数点后一位</p> <p>◇ 站点类别说明</p> <p>1 基本站 2 基准站 3 一般站 4 自动站</p> <p>✓ 文件内容举例: 2009 07 26 08 117 54511 116.4700 39.8000 55.0 6 1004.0 999999 24.0 19.0 270.0 1.0 1000.0 7.0 24.0 18.0 270.0 1.0 925.0 74.0 21.0 11.0 235.0 4.0 850.0 147.0 16.0 7.0 205.0 1.0 839.0 999999 16.0 6.0 999999 999999 700.0 309.0 4.0 1.0 15.0 6.0</p>
数据实体记录数	<p>每个文件大约 120 个站*20 层≈2400 行 注: 每个文件记录数不同, 每个探空站规定层固定 11 层, 特性层层数不定。</p>
数据在线服务要求	<p>4) 查询要求</p> <p>✓ 元数据检索条件要求:</p> <p>◇ 数据类型、数据子类、数据属性; (一级条件) ◇ 数据产生日期、选取取值范围 (二级条件)</p> <p>✓ 存储数据检索条件要求:</p> <p>◇ 数据类型、数据子类、数据属性; (一级条件) ◇ 数据产生日期、区域范围。</p> <ul style="list-style-type: none"> ● 区域包括: 省、华南、华中、江淮、长三角、华北、经纬度 ● 区域定义方式: 经纬度/站号 <p>5) 用户群 气象科研人员</p> <p>6) 权限要求 数据分为 2 个级别: 1 级用户: 在线注册简单审核, 可以查询 08、20 时次的国际交换数据; 2 级用户: 在线注册邮件审核, 可以查询全国所有探空站探空和加密探空资料。</p>
数据存放目录规则	<p>目录结构: base/high/yyyy</p>
特殊数据特性属性说明	<p>1. 缺测或无效值为 999999 2. 如果文件名时间与元数据时间和数据体时间不一致时, 则该文件不入库, 放入出错文件目录, 人工查错。</p>

数据分析情况, 制定数据存储方案, 包括磁盘容量要求、数据库软硬件要求, 以及数据文件目录划分层次图和数据库逻辑和物理设计。(几个逻辑库、表空间要求、分区依据和要求等)

2.4 灾害天气综合数据库系统功能性需求

由于灾害天气数据库系统建设项目负责制定卫星、雷达、探空、常规资料、数值预报、973 项目数据等气科院所需科研数据的存储管理策略, 根据策略对各类

观测数据进行统一管理，并以此为基础为气科院各级用户提供数据的在线查询和下载服务。因此，系统的两个鲜明特点——支持海量数据入库和提供准确无误的数据服务，成为本课题设计要达到的最重要要求。针对这两点要求，本课题构思系统的主要任务^[12]包括：

1. 建设合理的数据存储管理架构。对各类观测数据的数据格式、数据特性、数据量等进行全面的分析。根据数据特性，部分数据采用数据文件存储与元数据相结合的存储方式，将其数据内容解析入库。根据数据分析情况设计合理的文件系统存储目录及数据库存储架构，以便两者能够有机的结合起来为用户提供多元化的服务。建设合理的数据存储管理架构是整个项目成功实施的基础。
2. 历史数据预处理。结合气科院实际情况，对需要入库的各种历史数据文件制定合适的预处理策略；根据预处理策略，完成对历史数据的审核、判断、纠错、整合和推送，生成符合新系统数据格式要求的原始数据文件，保证系统的向下兼容性。
3. 设计数据存储管理服务。根据数据特点，对各类数据制定合适的数据存储管理策略；根据管理策略，自动化实现数据存储管理，包括提取元数据，按数据存储目录的设计规则存储数据文件，将部分数据内容解析入库，并完成数据编目。
4. 提供数据服务。提供用户及数据权限管理，在权限范围内，为各级用户提供元数据检索、数据检索、数据下载服务，并为特定用户提供周期性的特定接口的自动数据服务。
5. 系统运行管理。系统运行管理包括系统运行监控和系统参数配置管理。系统运行监控是为了实时监视系统运行情况，以便发生故障时能及时发现并处理。系统参数配置管理通过提供系统配置参数修改功能，便于系统的扩展和维护，提高系统的灵活性。

正如上述系统任务的描述，系统最重要的功能——数据存储服务分为两个部分。对于信息中心推送而来的实时数据，系统将直接执行正常的存储流程对其进行转存与数据入库。而对于系统实施前存储的历史数据，由于数据格式标准的变更，需要先由预处理机制对数据文件进行预处理，重新生成系统可以识别的标准格式

的文件再进行入库处理流程。

2.4.1 系统的参与者和目标

根据

2.2 灾害天气综合数据库系统需求获取模式所介绍的需求获取模式,我们确定了系统的参与者及其目标。参与者包括主要参与者与协助参与者,其表示如下图所示:

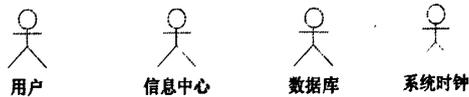


图 2-2 灾害天气综合数据库系统参与者图示

系统时钟是一个抽象的特殊参与者,这源于本系统需要在后台执行一系列定时自动数据处理操作,因此本文在描述这一类功能需求之时引入该参与者,利用其触发定时操作^{[13][14]}。

用户是一个高度抽象的参与者,是系统的使用者。通过执行系统的各项服务来完成日常工作。在灾害天气综合数据库系统中,我们考虑系统安全的需要,将用户分成多种角色,为每种角色赋予不同的用户权限。通过下面图示来简单表示系统的各种角色,并在图后进行各种角色权限的简单说明:

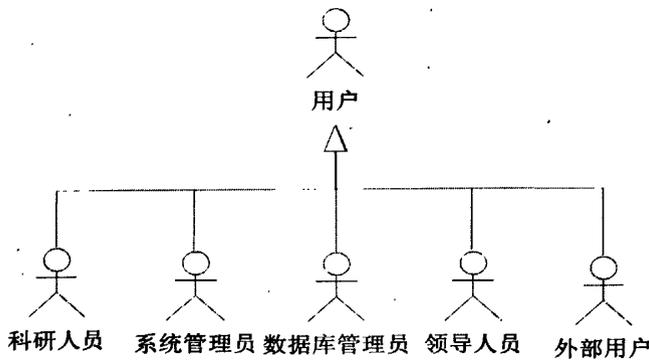


图 2-3 用户及角色权限图

系统管理员: 登陆管理页面,负责维护系统、配置系统配置表,进行角色管理,用户管理,日志管理,公告管理、建立数据访问规则、审核人员注册信息等日常维护性工作;

数据库管理员: 主要负责数据库日常维护工作,创建数据库存储结构

(tableplace)、创建数据库对象、修改和维护数据库的结构、数据库备份、灾害恢复等;

科研人员: 申请信息注册、设置查询条件查询灾害天气科研所需的各类气象数据、下载数据;

领导人员: 申请信息注册、设置查询各类气象数据、下载数据(权限高于一般的科研人员);

外部用户: 申请用户注册、设置查询条件查询气象数据、在有限领域内下载气象数据;

信息中心负责收集实时气象信息数据并推送给本系统使用。

数据库负责存储收集而来的各项要素数据, 并作为数据源为系统服务提供数据。

2.4.2 业务流程及用例

通过分析系统的参与者与角色的划分, 为方便对系统的理解, 本文将按照业务功能的划分, 结合本人实际的工作内容部分地整理系统的真实需求。

1. 数据存储需求调研

数据存储是灾害天气数据库系统建设的基石, 管理好数据存储, 首先要做好数据存储规划。存储规划是数据存储管理的前导性非实体功能, 其主要任务是在对本系统存储管理的所有数据的数据特性进行充分分析的基础上, 规划数据文件和数据库存储的物理位置和逻辑分布, 制定数据存储管理相关策略, 为数据存储管理的实施提供依据^[15]。以下给出系统所处理的数据特性统计分析方式:

数据特性分析是数据存储规划的首要工作, 只有在对各类数据的数据特性进行充分的分析归类, 才能对数据获取和存储策略进行有效的、合理的规划。

数据特性分析采用 2.3 数据分析模板介绍的数据调研特性卡的方式收集各类数据的数据特性, 在需求阶段主要完成以下几个方面的数据特性调查:

- (1) 数据源、数据到达频率及获取方式;
- (2) 数据文件命名规则;
- (3) 数据存储要求, 包括存储形态(文件或数据库)、存储期限等;
- (4) 元数据信息, 包括数据的元数据描述、关联文件信息、数据质量信息等

- (5) 数据文件内容格式说明;
- (6) 数据分类及分层原则;
- (7) 日数据量

以上每个数据特性的调查都将影响到下一步的规划设计。根据数据存储调研的结果,灾害天气数据库系统要求部分数据以文件形式存储,部分数据以数据记录方式存储在数据库中。以下给出具体的存储需求:

- (1) 以文件形式存储的数据有:卫星数据、雷达数据、常规资料数据、数值预报数据和项目数据(国家重点科研项目)等几大类数据;
- (2) 以数据记录方式存储在数据库中的数据有:常规资料数据,包括降水资料、探空资料、地面观测资料;
- (3) 对各类存储数据均要求提取元数据存储在数据库中用于数据检索;
- (4) 对以记录形式存储在数据库中的数据要求面向用户提供数据检索功能;

2. 数据预处理需求调研

灾害天气综合数据库不仅要处理国家气象信息中心推送而来的实时数据,实际上,根据气科院各科室要求,还要处理存储系统建设之前存储的历史数据文件。历史数据以文件形式存储在 GPFS 文件系统中,其提供的文件系统操作服务可以支持并行应用及串行应用,它允许任何节点上的并行应用同时访问同一个文件或者不同的文件,极大提高了文件的共享性^[16]。但由于一些历史原因,这些历史数据或是由于命名标准不统一,或是由于观测站、观测方式整合,并不能直接作为源数据文件参与本系统的数据存储流程。另一方面,由于观测站点涵盖全国各个地区,测站数量大,文件类型复杂,按照数据规划调研的结果,有一部分数据文件也需要从历史数据中抽取合并。综合上述原因,需要专门建立一个机制完成对数据文件预处理,生成系统可以识别处理的数据文件。由于对不同类型数据的预处理工作模式基本相似,下面以探空数据预处理为例说明数据预处理的需求:

用例 1: 数据预处理

参与者: 管理员

涉众及其关注点:

管理员: 希望能用最高效的方法对历史数据进行预处理,生成符合标准的

数据文件；

前置条件：待处理的历史数据文件存放在指定目录中；

后置条件：生成符合标准的数据文件，推送至系统的数据准备区；

主成功场景：

1. 管理员开始数据预处理工作；
2. 系统扫描历史数据存储区的目录，读取一个数据文件，根据文件名规则生成新数据文件；
3. 将区站号、经纬度等信息依次提取，将符合标准的数据条目按顺序写入新生成的数据文件；
4. 对新生成文件的测站数和观测层数进行统计，将结果写入文件；
5. 将新生成文件推送至系统存储准备区等待入库处理；

替代流程：

不符合标准的数据行剔除；

3. 数据存储管理需求调研

数据存储管理是灾害天气综合数据库系统两大核心实体业务之一，并且是另一核心业务（数据在线服务）顺利实施的基础。与存储规划一起作为本文的重点。它应根据规划好的数据存储策略实施数据存储流程，完成灾害天气综合数据库所管辖的所有数据的存储管理，为数据在线服务提供数据来源。

在通常的气象数据存储领域，数据入库依次要完成数据获取、元数据提取、要素数据解析入库和不合格数据处理等几大步骤。

数据获取的主要任务是根据各类数据的获取策略监视数据准备区相应目录，当有新的数据到达时，将数据移入数据存储区。

用例 2：数据获取

参与者：信息中心，系统时钟

涉众及其关注点：

信息中心：推送数据被灾害天气综合数据库系统接收、转存。

前置条件：系统获取参数配置表可用，对数据准备区的数据监视目录、数据到达时间间隔和数据储区的数据存储目录配置好参数信息；

后置条件：将文件从数据准备区移至数据存储区指定的目录下；将数据获取状态信息写入数据存储流程状态表中，并将正常日志信息记入数据库日志表中；

主成功场景：

1. 系统时钟调度系统进行数据获取，读取数据获取参数配置表，获取数据准备区的数据监视目录、数据到达时间间隔和数据存储区的数据存储目录等参数信息；
2. 根据目录监视时间要求，周期性扫描数据准备区各目录是否有新的数据送达；
3. 若有新的数据送达，则将文件从数据准备区移到数据存储区指定的目录下；
4. 数据获取成功，则将数据获取状态信息写入数据存储流程状态表中，并将正常日志信息记入数据库日志表中；

重复 2~4 步，直到当前目录为空

5. 每个目录数据获取完成后，进行下一个目录的数据扫描获取处理；
6. 若处理目录没有新数据送达，则直接进行下一个目录的数据扫描获取处理；

替代流程：

1. 读取参数配置表失败，则将失败信息写入数据库日志表中；
2. 数据获取失败，则将错误日志信息写入数据库日志表中；

特殊需求：数据准备区数据获取目录可用，数据存储区目录可用；

元数据提取是全部类型数据存储都要进行的工作。其主要任务是根据各类数据元数据的提取策略，从文件名或文件头(一般是文件内容首行)中取出相关元数据信息记入数据库。

用例 3：元数据提取

参与者：数据库，系统时钟；

涉众及其关注点：

系统时钟：调度系统从文件名或者文件头中提取出相关元数据存入数据库中。

前置条件：原始数据文件具有可读属性。

后置条件：各项元数据提取成功，写入元数据库；元数据与文件关联关系计入数据文件关联信息表；正常的日志信息记入数据库日志表；

主成功场景：

1. 系统时钟调度系统执行一次元数据提取工作，从数据存储流程状态表中检索某目录未提取元数据的数据文件名；
2. 系统依次从指定目录下读取各文件，根据元数据提取策略，对其文件名和文件头信息进行解析，获取相关元数据字段；
3. 组织提取出来的元数据信息，写入元数据库；
4. 将数据的关联关系记入数据文件关联信息表，并在数据存储流程状态表中置元数据提取成功标志；
5. 系统将元数据文件生成时间与最新数据信息表中的相关数据生成时比较，若时间比表中的时间大，则更新该数据；
6. 将子功能运行过程中产生的日志写入日志信息表；
7. 元数据提取完成后向调度员上报完成状态，等待进行下一步调度。

重复 2~7 步，直到数据存储流程状态表中该目录全部文件元数据提取完毕；

替代流程：

若元数据提取失败，则将此数据文件移到不合格产品区，并将此记录写入不合格数据表中，等待进行人工处理；然后在数据存储流程状态表中置元数据提取失败标志；

灾害天气综合数据库系统管理的数据中，一部分非结构化的数据文件以文件方式保存在外部存储器上，另外还有一部分结构化的数据文件需要将数据内容解析存储到数据库中，如常规资料。

用例 4：要素数据提取；

参与者：系统时钟，数据库；

涉众及其关注点：

系统时钟：调度系统读取结构化数据文件的内容，组织成数据记录写入数据库；

前置条件：数据文件可读；

后置条件：文件内容解析入库，写入相应的要素数据表；写数据存储流程状态表 and 对应元数据表；正常的日志信息记入数据库日志表；

主成功场景：

1. 系统时钟调度系统，根据数据获取参数配置表中的信息判断该类数据是否需要解析入库，如果需要则进入下一步；
2. 系统根据处理目录从元数据表中读取本目录未进行入库处理的文件名；
3. 系统根据读出的文件名依次读取、解析文件内容，组织各入库字段，将数据文件内容解析入库；
4. 系统在数据存储流程状态表和对应元数据表中置数据入库成功标志；
5. 将子功能运行过程中产生的日志写入日志信息表
6. 系统向调度员上报完成状态。

重复 1~6 步，直到全部需要解析入库的文件都被标记入库成功。

替代场景：

若解析入库失败，则将该文件移到不合格文件区，并将此记录写入不合格数据表中，等待进行人工处理；然后在数据存储流程状态表和对应元数据表中置数据入库失败标志；

不合格数据处理是数据存储管理对异常数据处理的补充，在正常存储流程中处理失败的数据被移到不合格数据区，由用户对其进行人工处理，这部分数据的处理情况分为两种，一种是删除，不再进行继续处理，还有一些数据用户进行修改后恢复正常，这些数据要求再次进行存储管理。

用例 5：不合格数据处理

参与者：系统管理员，系统时钟，数据库

涉众及其关注点：

系统管理员：对系统转存到不合格数据区的文件进行检查，可修复的数据文件入库，不可修复的数据文件删除；

前置条件：

不合格数据存储区非空；

后置条件:

不合格数据存储区为空;

主成功场景:

1. 系统管理员分析不合格数据区数据的具体情况,对有些无法修复或没有价值的数据进行删除,并在不合格数据表中置已删除状态;
2. 有些数据经修改后可继续使用,则管理员手动修改此数据文件;
3. 数据修改完成后,将数据文件移到存储区,并将记录插入到数据存储流程状态表中,重存标志置为“重存”;
4. 最后在不合格数据表中对应记录置重存状态;
5. 系统时钟调度系统进入数据存储流程;

于是数据存储管理用例总图可以绘制如下:

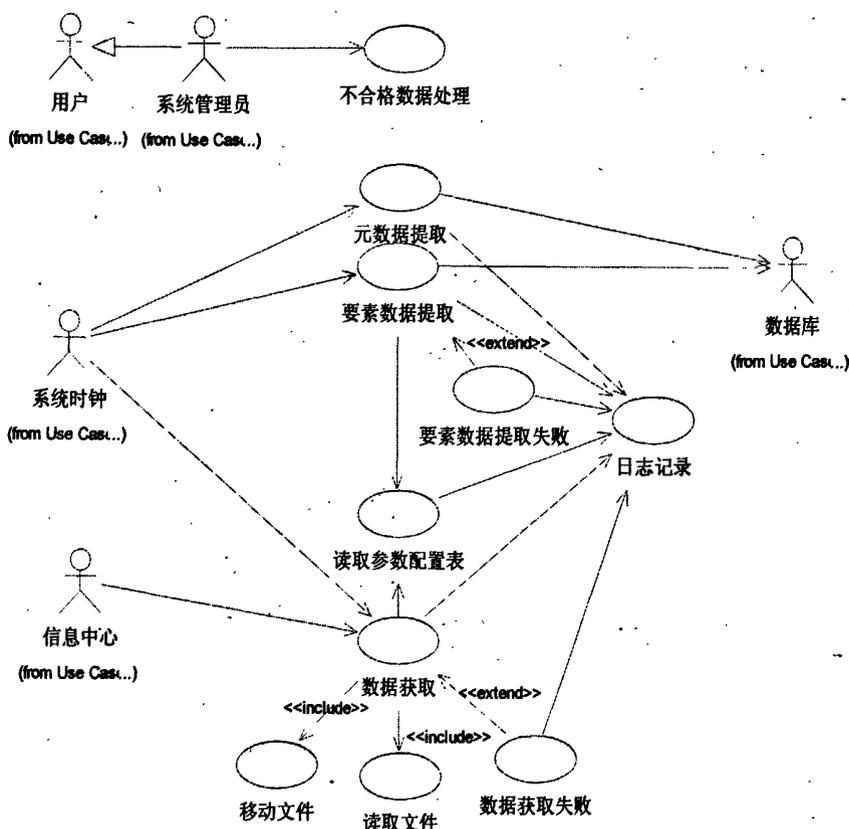


图 2-4 数据存储管理用例总图

2.5 灾害天气综合数据库系统非功能性要求

2.5.1 系统运行时效性

1. 数据存储管理响应时间：指系统将信息中心推送的源数据文件入库存档的最大相应时间。对于单个数据文件实体的存档最大时间小于 10 秒，元数据入库的时间小于 1 秒，对于接口区监控为不间断实时监控，扫描间隔为 5 分钟；
2. 数据服务信息响应时限：指用户通过系统界面查询、检索监控信息的最大响应时间。对于单个数据文件实体的检索时间小于 3 秒，数据下载最大时间小于 10 秒（由于数据文件大小不一样、实时网络带宽不一样，下载时间会有差异）；
3. 要求系统能够确保每类资料处理流程中的各个功能组件高效运行；
4. 要求系统整体具备负载均衡能力，可以根据资源的使用情况及数据处理、数据服务业务繁忙程度自动调节业务应用程序的运行环境和对资源的使用，对每个资料处理流程都能够实行并发运行。

2.5.2 可靠性要求

1. 成熟性标准：
7*24 小时业务运行期间，平均故障间隔时间（MTBF）不小于 1 个月。
2. 容错性标准：
软件接口句法性错误检出、屏蔽率 100%，其它软件错误的截获不超出直接传递控制的模块范围。
关键程序偶发故障恢复率 100%；
主机单点故障恢复率 100%；
3. 易恢复性标准：
平均故障恢复时间（MTTR）不大于 90 分钟。

2.5.3 维护性要求

1. 易分析性标准：
所有应用软件能够捕获到的计算机系统故障或应用软件错误都应在尽可能低

的级别上探查和捕获并以适当的形式记录或向操作员发送错误状态与报告；

错误状态和报告（或消息）应是统一编码、结构化和自描述的。即从错误状态码和错误消息正文串即可判断软件捕获到的错误的位置、类型和可能的原因；

错误报告（或消息）对于可预知（即可与程序的控制流程同步捕获到）的错误应能定位到独立的处理步骤或函数，不可预知（即不可与程序的控制流程同步捕获到）的错误应能定位到子功能或进程。

2. 易改变性标准：

关键任务及其任务环境参数可从外部配置和修改，并可从数据库中自动生成；

公用接口（包括错误状态码、常数等）结构应统一说明，所有对这些接口说明的引用都应当指向同一个说明实体；

3. 稳定性标准：

软件模块整体上应是结构化、层次化和功能单一的；

软件模块的处理和接口应是封装的。即处理的改变仅影响其所在的模块，接口的改变仅影响使用该接口的模块。

第3章 灾害天气综合数据库系统架构设计

灾害天气综合数据库系统满足中国气象科学研究院灾害天气数据的存储管理服务需求,实现对灾害天气科研相关气象数据的收集、加工、存储和处理生成的各类气象数据和产品的标准、规范化存储等管理功能,并提供满足共享服务需要的高效、规范的数据访问服务。

3.1 系统设计目标和原则

灾害天气综合数据库系统的设计目标主要体现在数据完整性、数据准确性、实时性、高效性、简单性、高可靠性、可扩展性和可维护性等几个方面^[17]。

第一,气象信息存储的完整性和准确性是系统首要关注的问题。灾害天气科学研究所涉及的数据全部来自全国各观测站的实测信息,是系统高效运行的前提和保障。系统之上所有服务的基础均来自气象信息的精确存储,因此系统必须做到数据的完整性和准确性;

第二,系统建成后,全部数据来源于信息中心的数据推送,数据源是实时的,因而系统也必须是实时的。对于推送而来种类繁多的实时数据,尤其是对于时次相近到达的规范化的入库数据,其综合内容堪称海量级别,这就在客观上要求本系统必须做到高效性。随着中心数据的不间断推送,这就对系统的实时性提出了很高的要求。

第三,系统应用主要面向的是气科院的科研人员,针对他们的关注点,在实现预定目标的同时,系统应该尽可能简单和易用。

第四,高可靠性要求也是系统设计的重要目标之一。系统实施后,软硬件运行过程中必然遭遇各种异常情况和突发事件。保证系统在这些干扰发生的同时可靠稳定地工作,才能保证系统的质量,体现系统价值。

第五,灾害天气综合数据库系统建设不仅限于气科院科研人员的日常应用,还对中国气象科研部门数据库系统建设起到示范作用。随着观测方式的不断革新,数据种类的不断增加,必然考虑今后系统功能的进一步增加。在最大程度上保证

系统的代码复用，降低系统与硬件之间、系统各模块之间的耦合度也就成为系统的一个重要目标。

高效的设计系统，必然引入设计约束和限制条件。为保证系统开发质量，本文分别在如下几方面进行约束：

1. 分析和设计策略：采用面向对象的分析和设计方法，RUP 作为软件开发过程，UML 作为建模语言；
2. 开发策略：分模块使用不同的开发工具，并按照职责和任务分工；
3. 技术规范：建立《灾害天气综合数据库系统建设项目数据分析文档》，参照执行《总装备部软件工程技术规范》
4. 平台规范：关系数据库采用Oracle 9i for Unix，开发工具使用MyEclipse5，开发包采用JDK 1.4.2 版本，服务器端操作系统采用IBM AIX Unix 5.1，客户端浏览器要求多浏览器支持，包括IE6.0及以上版本和FireFox等第三方浏览器。

3.2 灾害天气综合数据库系统技术架构设计

3.2.1 系统总体功能划分

由于系统主要负责对卫星、雷达、探空、常规资料、数值预报等科研数据的存储管理，并以此为基础向各级用户提供数据的在线查询和下载服务。为完成这些服务，结合需求分析所做工作，本文将灾害天气综合数据库系统分解为以下七个功能模块：

1. 数据存储规划功能；
2. 数据预处理功能；
3. 数据存储管理功能；
4. 数据在线服务功能；
5. 统一调度管理功能；
6. 系统监控及维护功能；
7. 信息管理服务功能；

这七个功能又分别分解成若干子功能，功能包如下图所示：

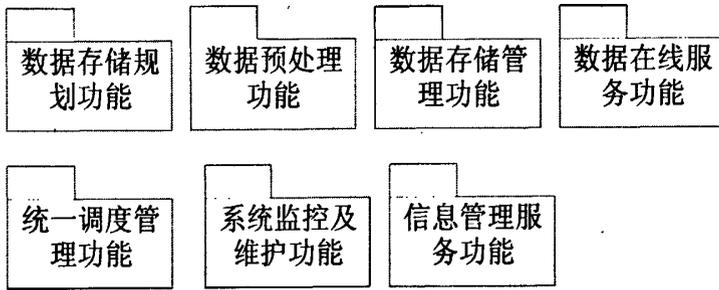


图 3-1 系统功能包划分图

数据存储规划应包括以下几项功能：数据特性分析，数据获取策略规划，数据文件存储规划，元数据提取策略规划，数据库存储规划；

数据存储管理功能包括数据获取，元数据提取，数据解析入库，不合格数据处理；

数据在线服务功能包括元数据检索，数据内容检索，数据应用展示，订单接收服务，数据下载服务，数据在线接口服务；

统一调度管理功能包括数据存储管理调度，数据在线服务调度，数据维护管理调度，信息管理调度；

系统监控及维护功能包括系统参数配置，数据存储状态监控，数据服务状态监控，数据库备份恢复，数据存储空间维护，日志信息监控等；

信息管理服务功能包括：用户注册，用户登录，最新数据发布，公告发布，栏目管理，快速链接，在线帮助等子功能。

图 3-2 显示了系统的总体功能结构

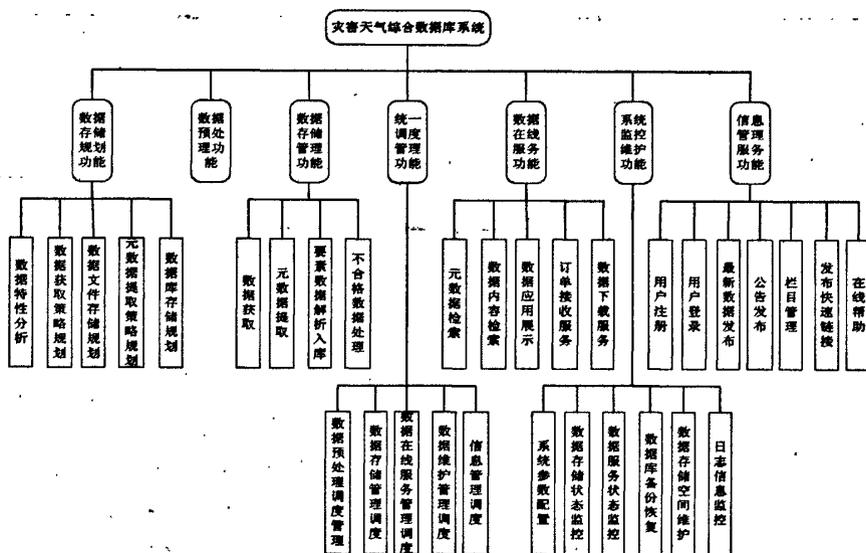


图 3-2 系统功能结构图

3.2.2 系统逻辑架构

从结构化角度来看，本系统要解决的首要问题既是目前气科院灾害天气数据管理中存在的主要问题，包括的数据完整准确入库保障；数据处理的时效性不能满足管理的需要；数据存储散乱，不利于系统统一结构化管理；数据共享程度低、无法提供在线数据服务、管理员无法对数据存储和在线服务实施集中化管理等。

为解决这些问题，理顺系统业务流程的各个环节，我们采用了对系统分层的思想，针对问题所在层面逐一进行设计。在系统的软件结构模式上，本系统采用时下最流行的 B/S 结构，使整个系统可以进行集中化管理和维护，无需进行客户端安装与维护操作。在逻辑上将系统分成了数据层、应用软件服务层、应用软件操作层、用户操作层，以及作为底层的软硬件支持层。采用多层体系结构可以将数据服务器与应用服务器进行多层部署，数据服务器可以集中部署在气科院数据中心内网，应用服务器则可以根据实际应用需求部署在数据中心内网或者外网。系统的逻辑视图参看图 3-3 灾害天气综合数据库系统逻辑视图。

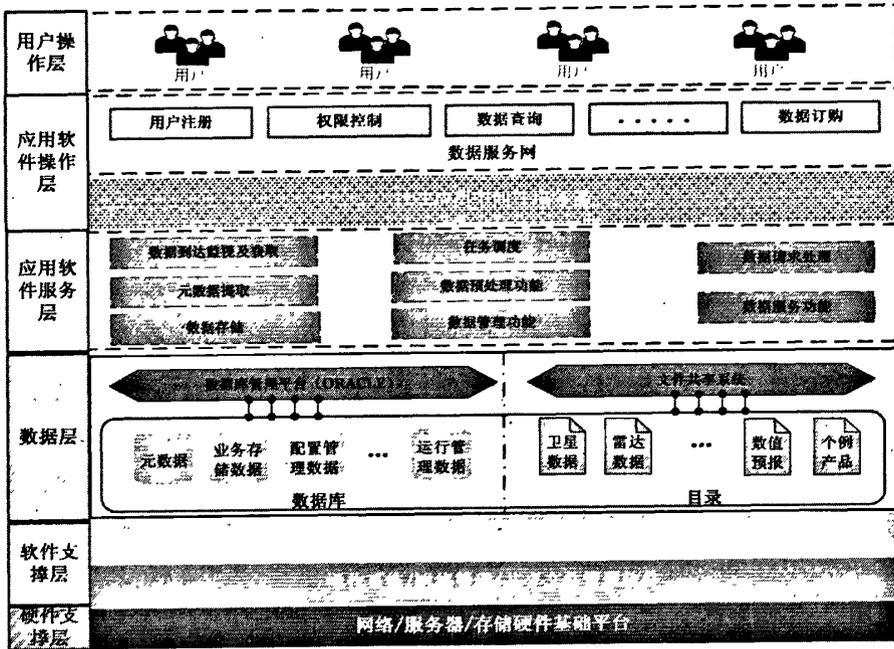


图 3-3 灾害天气综合数据库系统逻辑视图

3.2.3 系统功能流程

通过第 2 章需求分析的结果和 3.2.1 节对系统功能的划分，我们明确了灾害天气综合数据库系统前台后台各项应用的工作协作关系，用图 3-4 可以很好得表示这些业务流程。这里需要说明的是，由于数据存储规划属于系统非功能性业务，因此并没有绘制入系统业务流程图之中。

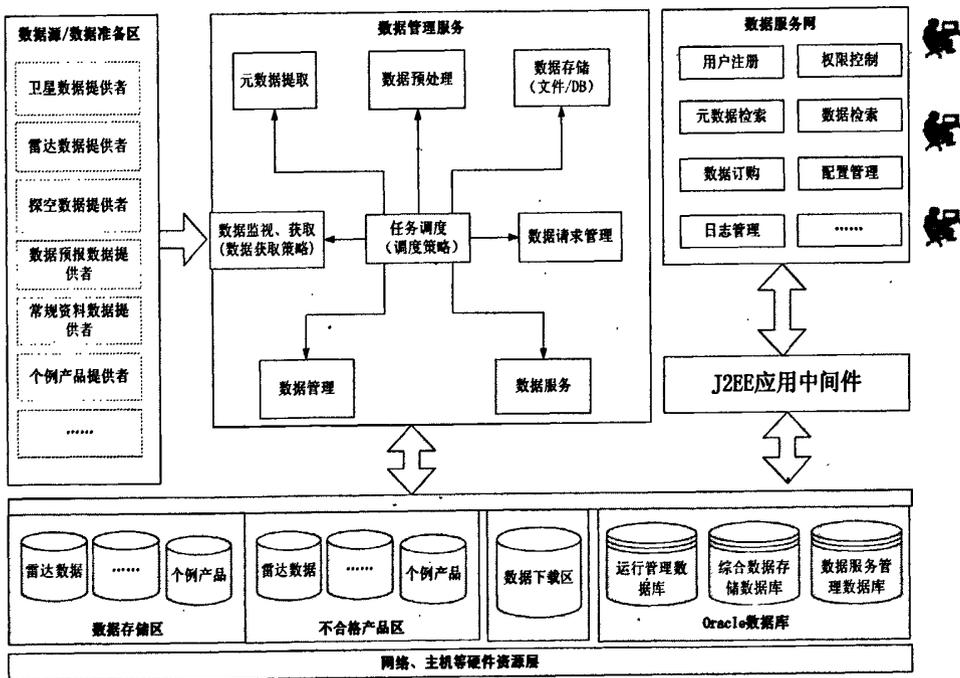


图 3-4 系统业务流程图

灾害天气综合数据库系统主体业务流程可以做如下简单陈述:

1. 系统管理员通过配置系统参数配置表设置数据获取方式和存储方式;
2. 中国气象信息中心推送各种气象数据至本系统数据准备区;
3. 任务调度策略调度系统数据存储管理模获取数据文件, 进行数据存储(包括文件存储和数据库入库存储);
4. 科研人员等数据用户通过客户端(浏览器)访问数据服务网, 进行数据检索、数据订单操作和数据下载操作; 数据服务网通过 J2EE 应用中间件访问数据库和数据下载区实现客户端的业务请求。

第 4 章 灾害天气综合数据库系统详细设计

通过第 2 章的需求分析和第 3 章架构设计,我们了解了灾害天气综合数据库系统的业务需求和架构流程。系统的重点在于存储,这是系统构建的基础,因此本章首先分析系统的存储规划设计。

4.1 灾害天气综合数据库系统存储规划设计

本文构建的灾害天气综合数据库系统,面向全国灾害天气信息数据的存储与管理,不但数据种类繁多,数据到达频率高,部分入库存储数据的内容也是惊人的。对这些海量数据实施合理的存储规划便成为系统高效运行的重要前提。通过需求中的调查,本文区分各类数据属性的共同点和不同点,对数据进行归类,对不同类型数据分别进行数据获取和数据存储策略的规划。上述各数据特性的调查结果,将作为各存储策略规划和设计的依据:

1. 数据获取策略规划的依据包括:数据源、数据到达频率及获取方式;
2. 元数据设计依据有:数据文件命名规则;元数据信息,包括数据的元数据描述、关联文件信息、数据质量信息等;数据文件内容格式说明中的头文件信息;
3. 数据存储策略规划依据(包括文件存储和数据库存储):数据存储要求,包括存储形态(文件存储或数据库存储)、存储期限等;数据文件内容格式说明;数据分类及分层原则;日数据量。

根据这些设计依据,本文给出规划过程中下面四步原则。

1. 数据获取策略规划

待数据分析工作完成后,确认数据的获取原则为是否采用目录监视,若是,确定各目录监视时间间隔,若不是则确定其它获取原则;

2. 数据文件存储规划

根据客户希望的数据存储方案列出每类数据的存储目录,及数据存储时限和空间回收要求;

3. 元数据提取策略规划

根据数据特性卡列出每类数据的元数据字段的基本要求、有无关联数据以及元数据字段提取方法等设定元数据的提取策略；

4. 数据库存储规划

列出每类要素数据所在逻辑库、表空间、分区，及数据存储时限和空间回收要求等确定数据库存储规划；

基于需求规定、设计思路以及技术体系架构等的设计约定，系统的数据存储管理主要包括两种存储形态，即基于文件模式的存储管理和基于数据库模式的存储管理。因此在存储规划设计中我们将针对文件和数据库两种存储管理模式进行分析和描述。

4.1.1 数据文件存储区域设计

基于文件模式的存储管理主要侧重于文件存储区的划分及文件存储区下文件目录结构层次的设计。

1. 存储区域划分

根据数据在灾害天气数据库系统中的数据流转过程，以及系统所管理数据的不同特性，将文件存储区域划分为数据准备区、数据存储区、数据下载区及不合格数据区四部分，其中数据存储区又分为基础资料数据区、项目资料数据区两部分。具体如下图所示：

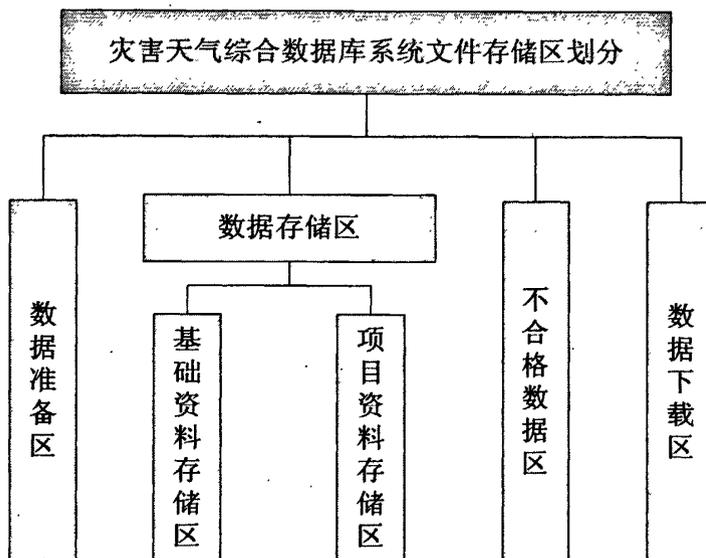


图 4-1 文件存储区划分图^[10]

(1) 数据准备区:

数据准备区是灾害天气综合数据库系统与中国气象信息中心的一个数据交换区域,它是灾害天气综合数据库系统数据流转的源头。数据提供方气象信息中心按照事先约定好的策略,按时主动将数据写入数据准备区,供灾害天气综合数据库系统使用。系统监视到有数据送达后即开始进入数据存储管理流程。

(2) 数据存储区:

一旦灾害天气综合数据库系统监测到数据准备区有新的数据送达后,便进入数据存储管理流程,该流程首先根据数据存储策略将准备区的数据移到数据存储区相应的存储目录下,然后对该数据进行元数据提取,元数据提取完成后,对于需要将文件内容解析入库的数据进行解析入库处理。

根据数据大类的不同的数据存储区又分为基础资料数据区和项目资料数据区,基础资料数据区存放的气象信息中心提供的所有基础资料数据,项目资料数据区存储的是气科院灾害天气国家重点实验室实验产生的个例研究数据,这两个数据区存储的具体内容包括:

基础资料数据区:卫星数据、雷达数据、常规资料数据、数值预报数据;

项目资料数据区:HIWs、973项目数据等;

(3) 不合格文件区:

在数据存储管理过程中,将元数据提取、验证出错或数据解析入库失败的数据文件移到不合格文件区,这些文件需要用户进行人工验证,修改成正确格式的数据后再次进入数据存储流程。

(4) 数据下载区:

为了保证数据存储区数据的安全,存放在该区的数据不直接为用户提供服务,当用户检索到所需要的数据,需要将数据下载到本地时,灾害天气数据库系统提供数据后台下载服务,将用户所需的数据拷贝到数据下载区,供用户下载使用。

2. 目录结构设计^[19]

根据数据用途的不同,数据文件存储区划分为四个不同的区域,各区域所存储的均不是单一类型的数据,为方便数据管理,根据数据管理策略对各数据区进行数据目录结构的设计,使数据存储有序、合理,便于查找使用。四个存储区的目录设计大致相同,篇幅所限,这里仅给出图 4-2 到 4-4 所示的数据存储区的目录

结构设计。

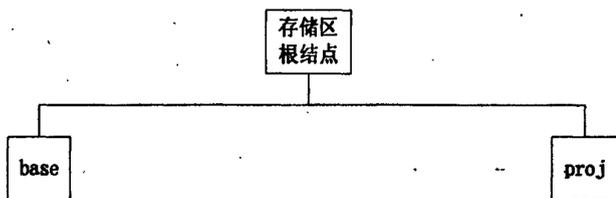


图 4-2 数据存储区顶级目录设计

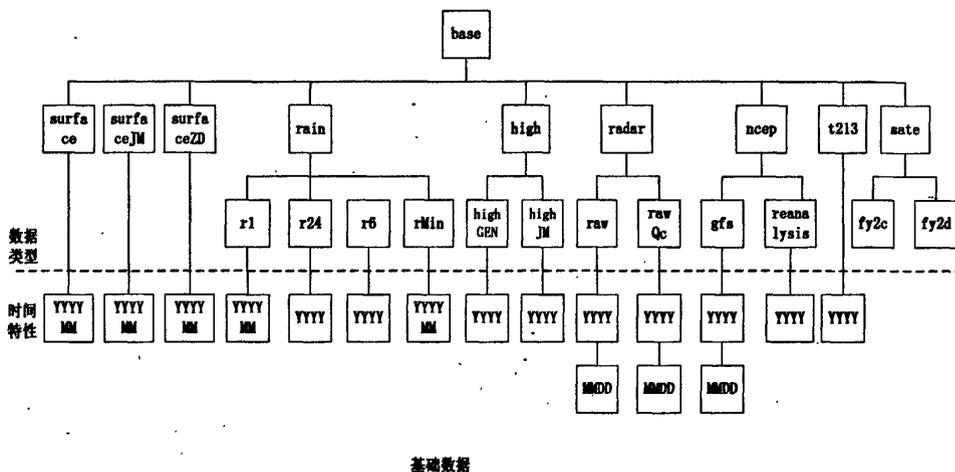


图 4-3 数据存储区基础数据存储目录设计

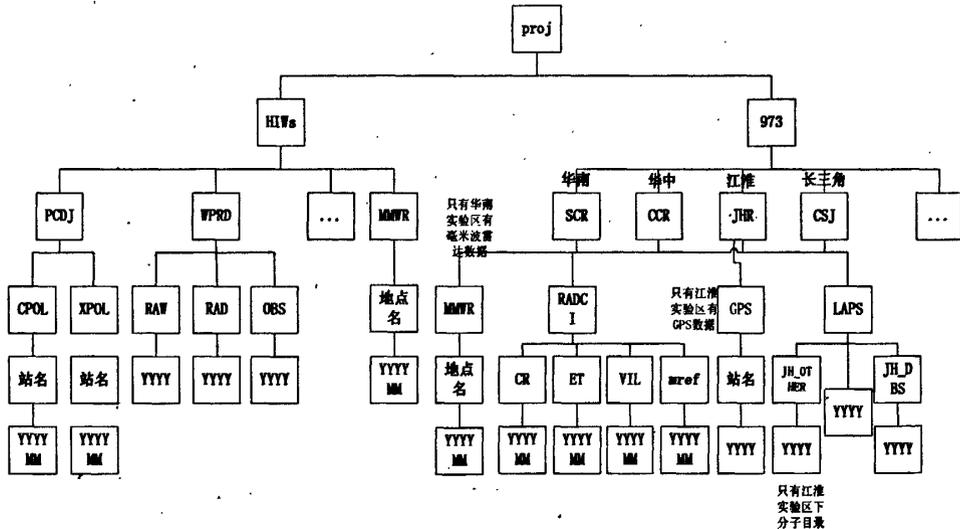


图 4-4 数据存储区项目数据目录设计

4.1.2 数据库逻辑模型设计

基于数据库模式存储管理的设计主要侧重于数据库信息模型的设计。同时无论以文件方式存储还是以数据库方式存储，都需要设计完整的、科学的元数据来描述数据的基本属性^[20]。

1. 设计原则

首先,在系统的数据库结构设计中我们以满足用户数据应用的实际需要为第一原则,同时将充分考虑应用的可扩展性,当数据应用发生变化时,数据模型将能快捷、简单支持数据应用的改变。

其次,系统运行效率和数据表规范相结合的原则。如必要时候可以通过数据表的部分冗余来提高运行效率和降低应用的开发复杂度。

2. 数据库表设计规范

(1) 公共字段的约定

静态元数据表及各类参数配置表都统一加上四个公共字段,即创建时间(CreateTime),创建人(CreateBy),最后修改时间(LastUpdateTime),最后修改人(LastUpdateBy)。

(2) 表命名规范

数据库表命名规则为: T_数据类型缩写_数据表名:

表 4-1 数据类型规则表

编号	业务数据类型名称	英文名称	英文缩写
1	元数据	MetaData	META
2	常规资料数据	Routine Data	ROUT
3	数据服务数据	Data Service	SERV
4	业务运行流程数据	Operation Flow	FLOW
5	参数配置数据	Parameter Configuration	PARA
6	页面管理数据	page setup	PAGE

数据表名含有一到多个单词,每个单词首字母大写,单词间不加分隔符;

每个数据库表名的长度不能超过 30 个字符;

数据库表字段命名规范:数据库字段名各英文单词的首字母大写,其他字母小写,单词间不加分隔符;字段名长度不能超过 30 个字符;

(3) 主键命名规范:主键名用 PK_开头,为 PK_表名_构成的字段名,主键名长度不能超过 30 个字符;

(4) 外键命名规范: 外键名用 FK_开头, 为 FK_外键表名_主键表名_外键字段名, 外键名长度不能超过 30 个字符, 其中的表名去掉其前缀, 如果过长, 可对表名进行缩写;

(5) 索引命名规范为: 前缀_表名缩写_索引字段缩写; 普通索引前缀为 IDX_; 主键索引前缀为 IDX_PK_; 唯一索引前缀为 IDX_UK_; 外键索引前缀为 IDX_FK_; 一般长度不超过 30 个字符;

3. 设计工具

在整个系统的数据库逻辑模型设计中, 将统一选择 PowerDesigner 作为数据库的建模工具。同时通过该工具实现整个数据库模型的维护与更新及生成整个数据库的建模脚本。

数据库设计中最典型和最常用到得是元数据库和常规资料数据库, 下面只给出这两类数据的模型设计。

4. 元数据库数据模型设计

由于数据库过于庞大, 本文只给出数据表清单和数据大类描述表结构说明

表 4-2 元数据库模型表

序号	表名	表说明
1.	T_META_DataClass	数据大类描述表
2.	T_META_DataType	数据类型描述表
3.	T_META_DataSubType	数据子类型描述表
4.	T_META_DataAttribute	数据属性描述表
5.	T_META_AreaCode	区域编码表
6.	T_META_ProjInfo	项目信息表
7.	T_META_RoutRadarArea	业务雷达站属性表
8.	T_META_RoutArea	常规资料探测站属性表
9.	T_META_ProjRadarArea	项目雷达站属性表
10.	T_META_ProjHighArea	项目探空站属性表
11.	T_META_ProjRoutArea	项目地面站属性表
12.	T_META_SateInfo	卫星信息表
13.	T_META_InstInfo	卫星仪器信息表

14.	T_META_DataLevel	数据级别信息表
15.	T_META_DataNorm	数据规格信息表
16.	T_META_DataFormat	数据格式信息表
17.	T_META_ProjectionMode	投影方式信息表
18.	T_META_Terminal	数据接收站信息表
19.	T_META_RoutineData	常规资料元数据表
20.	T_META_FileRelation	数据文件关联信息表
21.	T_META_NWPData	数值预报元数据表
22.	T_META_RadarData	雷达数据元数据表

表 4-3 数据大类描述表实体属性列表:

序号	名称	标识	数据类型	缺省值/值域说明	允许空	主外键
1.	数据大类标识	DataClassID	Char(8)		N	PK
2.	数据大类名称	DataClassName	Varchar2(50)		N	
3.	数据大类描述	ClassDesc	Varchar2(100)			
4.	创建时间	CreateTime	Date		N	
5.	创建人	CreateBy	Char(8)		N	
6.	最后修改时间	LastUpdateTim	Date		N	
7.	最后修改人	LastUpdateBy	Char(8)		N	

表 4-4 数据大类描述表数据实例

数据类型标识	数据类型名称	数据描述	创建/修改时间	创建/修改人
BASE	基础数据		YYYYMMDDHH24MIS	操作员或系统
PROJ	项目数据		YYYYMMDDHH24MIS	操作员或系统

5. 常规资料数据数据库模型

表 4-5 常规资料数据库表清单

序号	表名	表说明
1.	T_ROUT_Rain1Minute	常规资料逐分钟降雨资料数据表
2.	T_ROUT_Rain1Hour	常规资料逐时降雨资料数据表
3.	T_ROUT_Rain24Hour	常规资料 24 小时降雨资料数据表
4.	T_ROUT_Rain6Hour	常规资料 6 小时降雨资料数据表
5.	T_ROUT_SurfaceAuto	常规资料地面自动站资料数据表

6.	T_ROUT_SurfaceGeneral	常规资料地面常规观测资料数据表
7.	T_ROUT_High	常规资料探空与探空加密资料数据表
8.	T_ROUT_SurfaceJm	常规资料地面常规资料加密数据数据表

表 4-6 以探空资料数据表为例给出常规资料表结构说明

	名 称	标识	数据类型	缺省	允许空	主外键
1.	文件名	FileName	Varchar2(100)		N	PK
2.	行号	LineNo	Int		N	
3.	数据产生时间	ProduceDate	Date		N	
4.	数据产生年	ProdYear	Char(4)		N	
5.	数据产生月	ProdMonth	Char(2)		N	
6.	数据产生日	ProdDay	Char(2)		N	
7.	数据产生时	ProdHour	Char(2)		N	
8.	探空站数	StationCount	Int		N	
9.	区站号	StationNum	varchar2(10)		N	PK
10.	转换后区站号	TranStationNum	Varchar2(10)			
11.	标准站点类别	StationType	int		N	
12.	经度	Longitude	Numeric(9,4)		N	PK
13.	纬度	Latitude	Numeric(9,4)		N	PK
14.	海拔高度	SeaLevelHeight	Numeric(8,1)		N	
15.	总层数	TotalLevel	int		N	
16.	层数	LevelNum	int		N	
17.	气压	Pressure	Numeric(8,1)		N	
18.	高度	Height	Numeric(8,1)		N	
19.	温度	Temperature	Numeric(8,1)	整数	N	
20.	露点	DewPointTemp	Numeric(8,1)	整数	N	
21.	风向	Wind	Numeric(8,1)		N	
22.	风速	WindSpeed	Numeric(8,1)	整数	N	
23.	气压质量码	PressureQc	Int	整数		
24.	高度质量码	SeaLevelHeightQc	int	整数		
25.	温度质量码	TemperatureQc	Int	整数		
26.	露点质量码	DewPointTempQc	Int	整数		
27.	风向质量码	WindQc	Int	整数		
28.	风速质量码	WindSpeedQc	Int	整数		

6. 数据库优化策略

灾害天气综合数据库系统涉及海量数据存储,以逐分钟降水数据为例,根据数据调研结果,每天将有 24 个文件推送存储(间隔时间为一小时),每个文件涉及全国 20000 多个雨量观测站,一年的数据就多达 2 亿条之巨。数据库优化便成为设计时必须考虑的一大问题。此外,由于数据库表众多,数据查询与插入耗时将直接影响系统整体性能。本文进行设计时引入 oracle 分区功能,以改善关系数据库的可管理性、性能和可用性。通过分区功能,数据库将表、索引进一步细分为段,从而可以更精确和快速地访问这些数据库对象。这样的分区,对 SQL 语句是完全透明的,因而并不会增加开发的难度。为此本文专门进行了数据库的进行对比测试,下图是两百万条记录以内数据库检索耗时对比图。

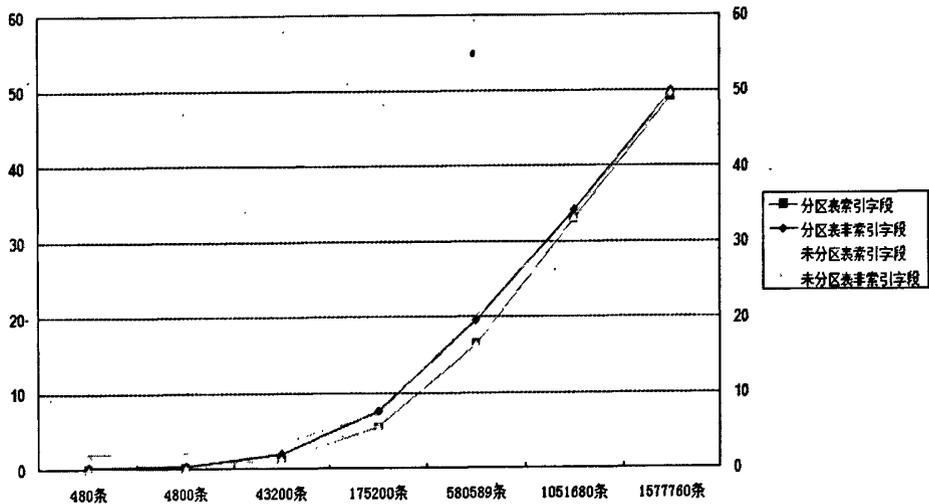


图 4-5 两百万级别检索耗时对比图

图中显示分区表的查询性能一直有比较好的表现,然而图中的两百万级别数据查询耗时结果显示,建立索引字段不当,将严重影响数据库的性能。在进行配置排查和查阅相关资料后,我们了解到索引字段过多将增加索引快速全扫描的成本,进而影响数据检索的性能^[21],因此本文确定以一般数据库表不超过 4 个索引字段为原则进行索引设置和主外键设置。

4.2 系统建模

4.2.1 系统静态建模

通过上面的数据存储规划设计，本文确定了系统实现的数据存储方式和数据库表结构。结合业务需求和系统架构，本文在这一节对系统功能和结构进行细化。在设计时，按照高内聚，低耦合的设计原则，遵从公司的公用架构，将系统分作五个包。这五个包分别是基础包(公共服务包)、模型包、存储解析包、数据服务包和系统服务包。下面是这五个包的说明：

1. 基础类包：主要包含通用的公共模型基础类和公共服务基础类；
2. 模型类包：定义系统数据存储解析和数据服务操作对象的类；
3. 存储解析类包：包含全部存储解析类；
4. 数据服务类包：包含系统数据服务相关类；
5. 系统服务类包：包含系统配置及维护相关类；

系统类包及其相互关系如图 4-6 所示。

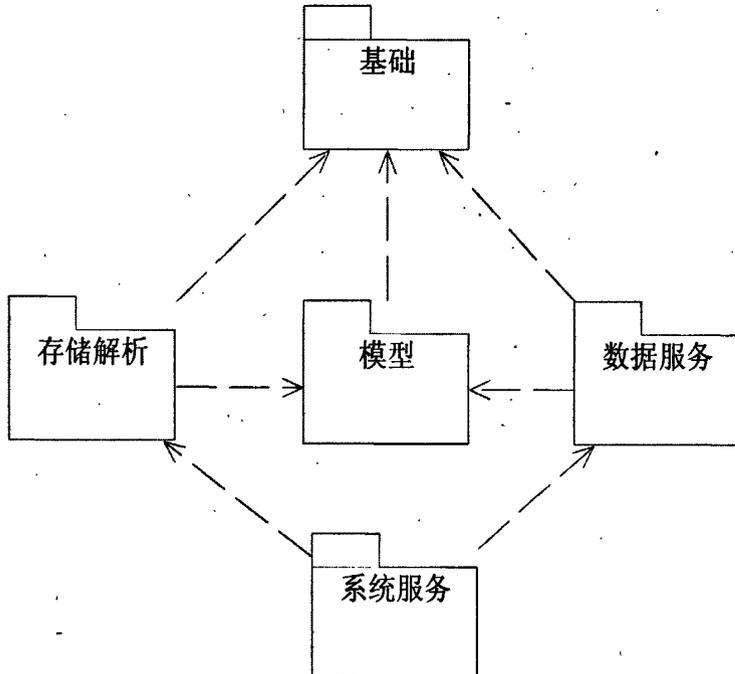


图 4-6 系统类包关系图

上面介绍了系统类包及其相互关系，如果展开细述，篇幅过长，所以本文仅对核心业务相关类及其关系进行介绍。本系统是建立在综合型气象数据存储之上

的数据库系统，数据服务等系统功能围绕数据存储才能展开。因此数据存储服务是系统最重要功能，是系统的核心。系统数据存储的业务实体在数据库中表现为关系数据，对数据库的存储和检索操作最终都是对关系数据的操作。因而在设计时本文使用了关系数据模型，通过创建模型的实体对象，执行对各类数据表的操作。由于本文涉及到 51 个气象数据表，其中实体业务数据表为 39 个，下面以常规资料为例，说明数据存储解析服务类之间的相互关系。

1. 数据实体模型类

对常规资料数据，由于数据存储方式分为文件存储与数据库存储，数据实体对应参与元数据存储与要素数据存储，因而在存储流程中，分别对应设计了 1 个元数据模型类与 8 个分项数据模型类，以及 1 个参数配置表类。这些类分别是

TRouteRuntimeData:常规资料元数据类，对应常规资料元数据表；

TRouteHigh:常规资料探空要素数据类，对应常规资料探空数据表；

TRouteRain1minute:常规资料逐分钟降雨要素数据类，对应常规资料逐分钟降雨数据表；

TRouteRain1hour:常规资料逐时降雨要素数据类，对应常规资料逐时降雨数据表；

TRouteRain6hour:常规资料 6 小时降雨要素数据类，对应常规资料 6 小时降雨数据表；

TRouteRain24hour:常规资料 24 小时降雨要素数据类，对应常规资料 24 小时降雨数据表；

TRouteSurfaceauto:常规资料地面自动站要素数据类，对应地面自动站数据表；

TRouteSurfacegeneral:常规资料地面常规观测要素数据类，对应常规资料地面常规观测资料数据表；

TRouteSurfacejm:常规资料地面加密资料要素数据类，对应地面常规资料加密数据数据表；

TParaGetData:参数配置表类，对应参数配置表；

为方便查询，我们在设计数据库时为各数据表设置了一定的索引项。在设计实体模型时这些索引项被设计为相应的 Id 类，以提高数据存储及数据服务检索效率。

下面以常规资料探空数据类为例给出常规资料要素数据类属性及方法的详细说明:

【类名称】: TRoutHigh 类

【类描述】: 主要对应探空数据的解析、删除、查询操作的实体对象。

【属性列表】

表 4-7 常规探空资料要素数据类属性表

序号	属性	属性名称	属性说明
1	Id	主键	为该类的私有属性, 标志探空数据的主键
2	lineno	数据所在行	标志本条探空数据所在源数据文件中的行数
3	producedate	生成日期	标志探空数据的生成日期
4	prodyear	生成年份	标志探空数据的生成年份
5	prodmonth	生成月份	标志探空数据的生成月份
6	prodday	生成日	标志探空数据的生成日
7	prodhour	生成时刻	标志探空数据的生成时刻
8	stationcount	记站数	标志探空数据源数据文件中测站数目
9	transtationnum	测站号	标志本条探空数据的测站站点号
10	stationtype	测站类型	标志探空数据测站类型
11	longitude	测站经度	标志测站的经度(东经)
12	latitude	测站纬度	标志测站的纬度(北纬)
13	sealevelheight	海拔高度	标志测站的海拔高度
14	totallevel	站点总层数	标志本条探空数据记录的站点总层数
15	pressure	当前层气压	标志当前层的大气压力
16	height	当前层高度	标志当前层的海拔高度
17	temperature	当前层温度	标志当前层的温度(摄氏度)
18	dewpointtemp	当前层露点	标志当前层的露点温度
19	wind	当前层风向	标志当前层的风向
20	windspeed	当前层风速	标志当前层的风速
21	pressureqc	气压质量码	标志当前层气压的质量码(各要素质量码雷同, 略)

【方法列表】

表 4-8 常规探空资料要素数据类方法表

序号	方法	方法名称	方法说明
1	TRoutHigh	构造方法	公用方法, 该方法的主要作用是构造探空数据类对象
2	getId/setId	读取/设置 Id	公用方法(下同), 该方法的主要作用读/设置 Id。
3	getLineno/setLineno	读取/设置行数	该方法的主要作用是读取/设置本条记录所在源文件中的行数
4	getProducedate/setProducedate	读取/设置生成日期	该方法的主要作用读取/设置探空数据的生成日期。
5	getProdyear/setProdyear	读取/设置生成年份	该方法的主要作用是读取/设置探空数据的生成年份。
6	getProdmonth/setProdmonth	读取/设置生成月份	该方法的主要作用是读取/设置探空数据的生成月份。
7	getProdday/setProdday	读取/设置生成日	该方法的主要作用是读取/设置探空数据的生成日。
8	getProdhour/setProdhour	读取/设置生成时刻	该方法用作读取/设置探空数据的生成时刻。
9	getStationcount/setStationcount	读取/设置测站数	该方法用作读取/设置探空数据源文件包含的测站数目。
10	getTranstationnum/setTranstationnum	读取/设置转换后测站号	该方法用作读取/设置探空数据源文件包含的测站数目。
11	getStationtype/setStationtype	读取/设置站点类型	该方法用作读取/设置探空数据测站的站点类型。
12	getLongitude/setLongitude	读取/设置站点经度	该方法用作读取/设置探空数据的站点所在经度。
13	getLatitude/setLatitude	读取/设置站点纬度	该方法用作读取/设置探空数据的站点所在纬度。
14	getSealevelheight/setSealevelheight	读取/设置海拔高度	该方法用作读取/设置探空数据的站点所在海拔高度。
15	getTotallevel/setTotallevel	读取/设置总层数	该方法用作读取/设置探空数据的站点观测的总层数。
16	getPressure/setPressure	读取/设置当前层气压值	该方法用作读取/设置探空数据的所在层的气压值。
17	getHeight/setHeight	读取/设置当前层海拔高度	该方法用作读取/设置探空数据的所在层的海拔高度。
18	getTemperature/setTemperature	读取/设置当前层的气温值	该方法用作读取/设置探空数据的所在层的气温值。
19	getDewpointtemp/setDewpointtemp	读取/设置当前层露点	该方法用作读取/设置探空数据的所在层的露点温度值。
20	getWind/setWind	读取/设置当前层风向	该方法用作读取/设置探空数据的所在层的风向。
21	getWindspeed/setWindspeed	读取/设置当前层风速	该方法用作读取/设置探空数据的所在层的风速值。
22	getQc/setQc	读取/设置各要素数据质量码	该方法用作读取/设置探空数据各要素数据的质量码。

2. 存储解析处理类

本系统在进行数据存储服务时,是按照数据获取——获取文件参数配置信息——文件元数据解析——文件要素数据解析(部分)——存储的核心流程进行处理。于是在设计阶段我们将其主要设计为两个类:数据收集类 DataCollectionMgr 及数据解析类 ParseMgr 类。下面给出这两个类的详细说明:

(1) 数据解析管理类

【类名称】: ParseMgr 类

【类描述】: 主要进行全部灾害天气气象数据的元数据要素数据解析操作。

【主要方法列表】

表 4-9 数据解析管理类方法列表

序号	方法	方法名称	方法说明
1	getFileheadlineno	获取文件头行号	静态方法, 获取文件头所在行号
2	parseMetaRoutineData	解析元数据	静态方法, 统一对各类常规数据文件名所包含的元数据进行解析
3	parseElementROUT_High	解析探空资料要素	静态方法, 对探空数据文件的要素数据进行解析
4	parseElementROUT_Rain1minute	解析逐分钟降雨资料要素	静态方法, 对逐分钟降雨数据文件的要素数据进行解析
5	parseElementROUT_Rain1hour	解析逐时降雨资料要素	静态方法, 对逐时降雨数据文件的要素数据进行解析
6	parseElementROUT_Rain6hour	解析 6 小时降雨资料要素	静态方法, 对 6 小时降雨数据文件的要素数据进行解析
7	parseElementROUT_Rain24hour	解析 24 小时降雨资料要素	静态方法, 对 24 小时降雨数据文件的要素数据进行解析
8	parseElementROUT_SurfaceAuto	解析地面自动站要素	静态方法, 对地面自动站数据文件的要素数据进行解析
9	parseElementROUT_Surfacegeneral	解析地面常规观测要素	静态方法, 对地面常规观测站数据文件的要素数据进行解析
10	parseElementROUT_Surfacejm	解析地面观测加密数据要素	静态方法, 对地面站加密数据文件的要素数据进行解析
11	parseMetaProj973	解析 973 项目数据元数据	静态方法, 对“973 项目”数据元数据进行解析

(2) 数据收集管理类

【类名称】: DataCollectionMgr 类

【类描述】: 主要进行全部灾害天气气象数据进行收集和存储操作。

【主要方法列表】

表 4-10 数据收集管理类方法列表

序号	方法	方法名称	方法说明
1	processTask	任务处理调度	公共方法, 调度执行数据入库流程
2	moveFile	源文件移动	静态方法, 移动文件
3	funcCollection	收集文件	公共方法, 按照处理流程对数据文件进行收集
4	funcParse	解析方法	公共方法, 对收集文件的元数据进行解析
5	funcEle	解析要素	公共方法, 对所文件的要素数据进行解析
6	funcSave	数据保存	保存的元数据和要素数据
7	parseMeta	解析元数据方法	公共方法, 按照工厂模式分发元数据解析任务
8	parseMetaRoutineData	解析常规资料元数据	公共方法, 解析常规资料元数据
9	parseElements	解析要素数据	公共方法, 对数据文件的要素数据进行解析
10	parseElementROUT_High	解析探空要素	公共方法, 对探空数据文件的要素数据进行解析
11	insertRoutHigh	向数据库插入探空数据	公共方法, 解析所得探空数据进行入库操作
12	getautoHighSql	获取探空数据插入SQL	公共方法, 获取探空数据入库的 SQL 语句

于是系统核心业务——数据存储中各类之间的关系如图 4-7 所示。

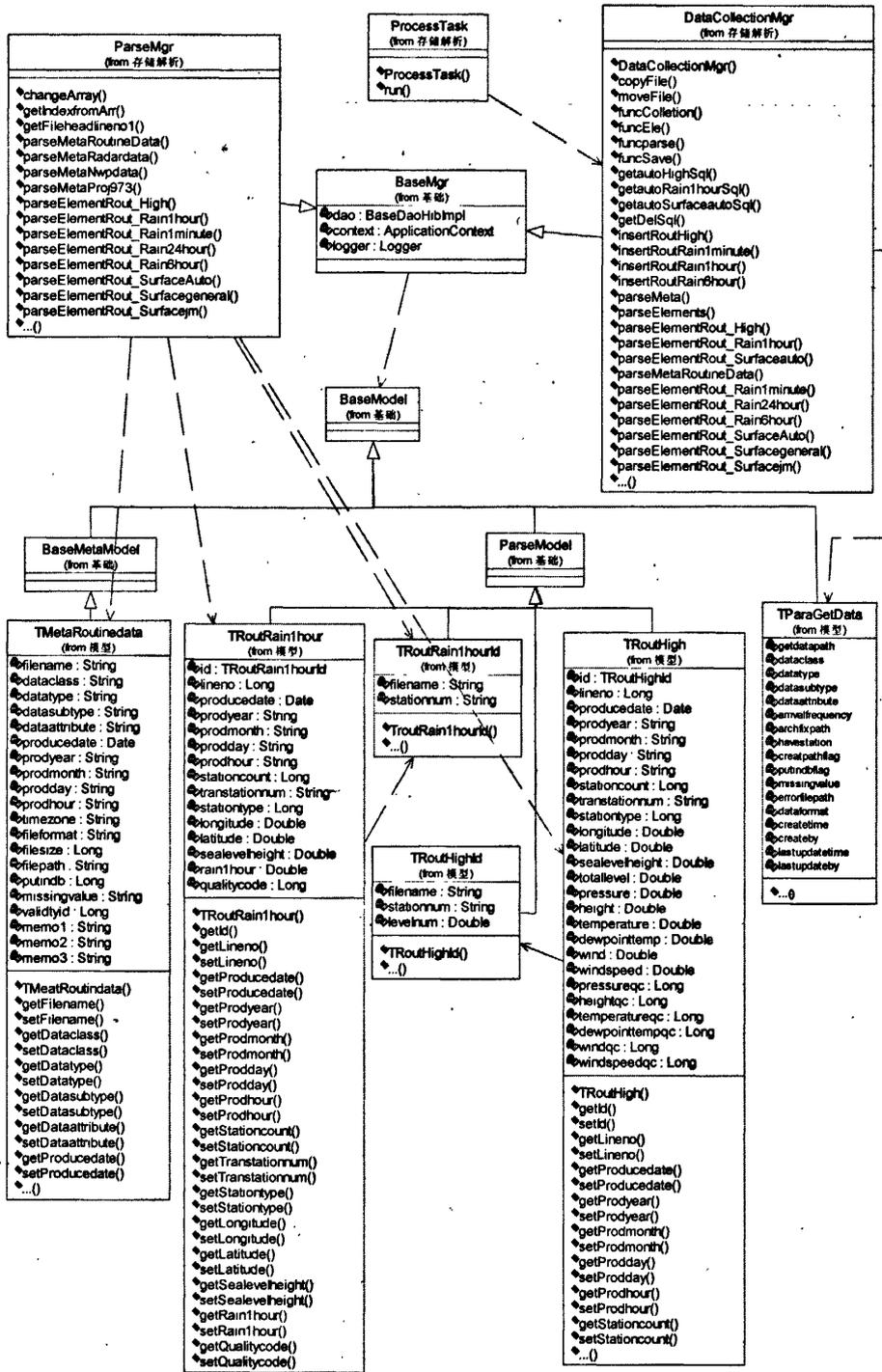


图 4-7 系统核心业务类图

4.2.2 系统动态建模

仍以数据存储服务为例，对于定时推送而来的数据文件，我们设计的数据存储流程可以参看如图 4-8 所示的时序图。

灾害天气气象数据信息经过如下几个步完成数据入库操作：

1. 信息中心推送数据至数据准备区；
2. 系统时钟定时启动数据扫描与存储任务；
3. 系统扫描数据准备区，对准备区存放的每一个气象数据文件进行扫描，将符合存储标准的文件转存至文件存储区；
4. 系统根据参数配置表，获取待入库文件的存储要求；
5. 根据存储要求对源数据文件进行元数据解析，并对部分数据文件进行要素数据解析；
6. 将解析结果以数据记录的形式存入相应的数据库表。

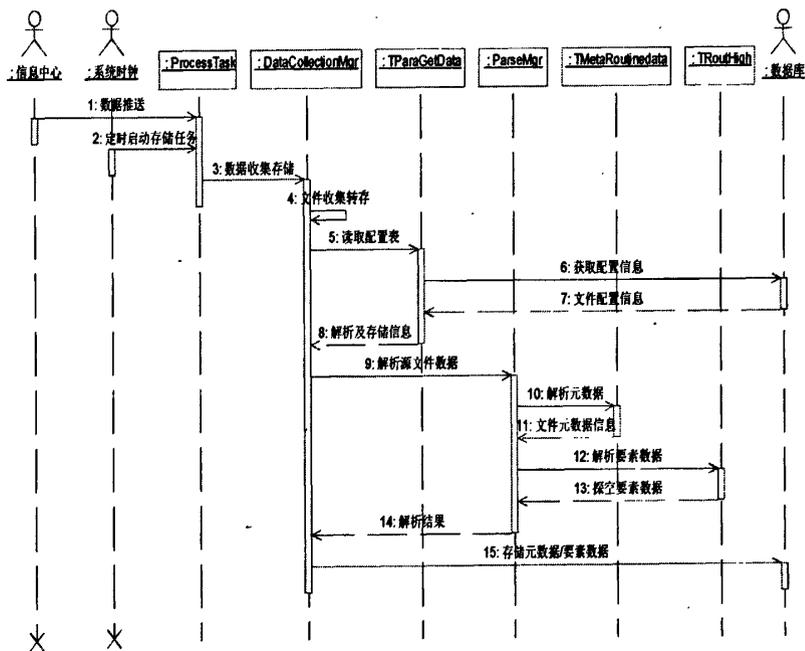


图 4-8 数据存储解析时序图

为了更好的表述系统进行存储服务时各对象之间的协作关系，本文用如图 4-9 所示的协作图来说明各对象彼此之间的链接。

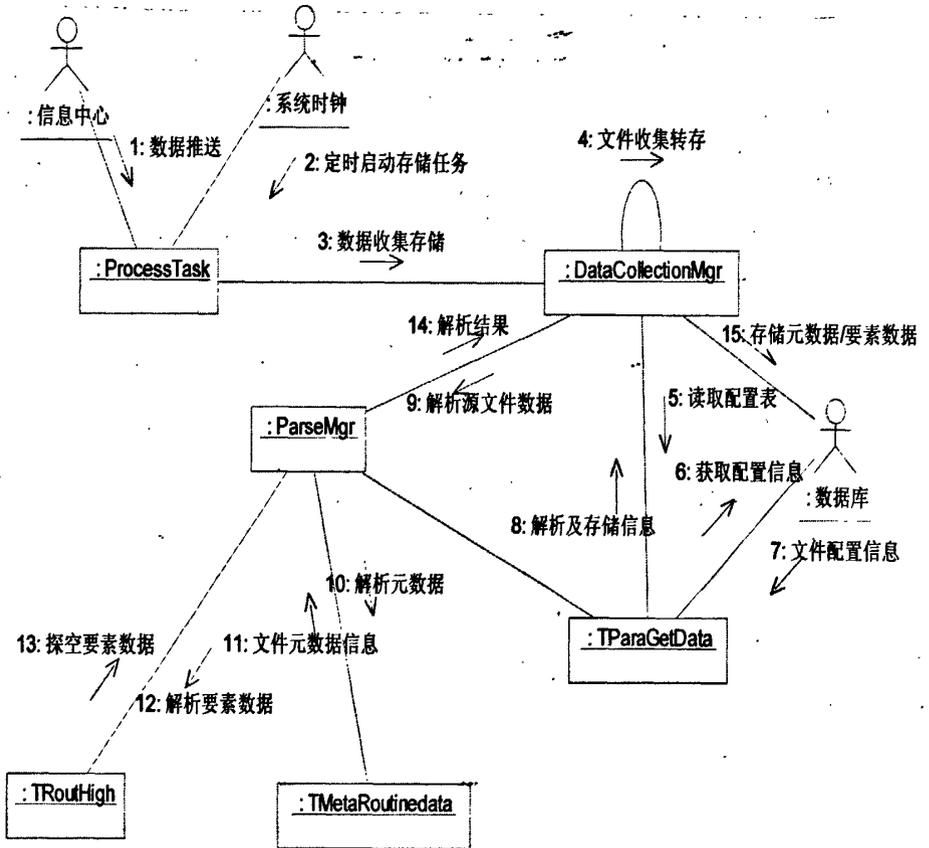


图 4-9 数据存储协作图

第 5 章 灾害天气综合数据库系统实现

系统实现阶段的主要工作是面向对象的程序设计,在公司成熟的代码框架的基础上对数据预处理功能、数据存储管理功能、数据在线服务功能、统一调度管理功能、系统监控维护功能和信息管理服务功能等进行代码实现。本章首先给出灾害天气综合数据库系统的前台界面实现,结合开发所使用的公司框架说明系统运行的配置,再根据本人的工作重点对后台的数据预处理和数据存储管理功能的实现说明。

5.1 系统的总体界面与配置

综合前面的分析与设计,系统前台界面实现如下图所示。



图 5-1 系统前台界面

由于系统的实现采用了公司内部架构,这里只给出部分配置信息,从总体上说明系统各功能是如何配置并协作运行的。

公司的内部架构采用了 Spring+Struts+Hibernate+Quartz 等多项技术。其中后台的调度采用了 Spring 和开源的 Quartz 框架,在 Struts 的核心 struts-config.xml 中配置如下信息,便将 Spring 和 Quartz 配置完成。

```

<plug-in className="com.dhcc.common.ContextLoaderPlugIn">
  <set-property property="contextConfigLocation"
    value="/WEB-INF/base_applicationContext.xml,
      /WEB-INF/dc/dc_applicationContext.xml,
      /WEB-INF/dc/dc_action-servlet.xml,
      /WEB-INF/ds/ds_applicationContext.xml,
      /WEB-INF/ds/ds_action-servlet.xml,
      /WEB-INF/sys/sys_applicationContext.xml,
      /WEB-INF/sys/sys_action-servlet.xml,
      /WEB-INF/dc/quartz_applicationContext.xml
    "/>
</plug-in>

```

在系统的存储流程中，数据获取是第一步。本文的开发框架中使用 Quartz 管理后台调度，监视数据准备区目录，在上面代码引述的 quartz_applicationContext.xml 中配置触发器^[22]，定时扫描数据准备区的目录。下面的配置信息以常规资料探空数据为例给出配置：

```

<bean id="Trigger_BASE_ROUT_HIGH_GEN"
class="org.springframework.scheduling.quartz.SimpleTriggerBean">
  <property name="jobDetail">
    <ref bean="qd_BASE_ROUT_HIGH_GEN"/>
  </property>
  <property name="startDelay">
    <value>1000</value>
  </property>
  <property name="repeatInterval">
    <value>86400000</value>
  </property>
</bean>
<bean name="qd_BASE_ROUT_HIGH_GEN"
class="org.springframework.scheduling.quartz.JobDetailBean">
  <property name="jobClass">
    <value>com.dhcc.dc.business.DcQuartz</value>
  </property>
  <property name="jobDataAsMap">
    <map>
      <entry key="dataclass">
        <value>BASE</value>
      </entry>
      <entry key="datatype">
        <value>ROUT</value>
      </entry>
      <entry key="datasubtype">
        <value>HIGH</value>
      </entry>
      <entry key="dataattr">
        <value>GEN</value>
      </entry>
    </map>
  </property>
</bean>

```

需求中我们知道,常规探空资料的到达频率为每日 2 个文件,因此配置中将触发器设置为启动间隔频率 24 小时。

5.2 数据预处理功能的实现

数据预处理功能根据配置参数对不同气象数据进行不同的预处理,由文件名检查/修正、文件格式/内容提取、新文件生成和预处理日志记录功能构成。

5.2.1 数据预处理功能信息流程

数据预处理功能是根据气象数据传输规范,对历史文件的文件名、文件头、文件格式和文件内容进行标准化处理,生成符合系统要求的标准化数据文件推送至数据准备区。

数据预处理分系统的功能流程如图 5-2 所示。

5.2.2 数据预处理功能实现

1. 文件名检查/修正

根据气象信息标准规范,对历史文件的文件名进行正确性检查,检查的内容包括将文件名格式检查、文件名字符检查等。经过正确性检查后,将文件名与文件头(一般是文件的第一行)进行比对,文件名与文件头信息相符的文件可以写入新文件,不正确的文件将作为垃圾文件送至不合格文件区。

2. 文件格式/内容提取

对于历史气象数据文件,根据标准规范定义进行相应的文件格式检查。符合格式要求的文件,进行文件内容的检查与抽取,对要素数据缺测的数据行进行剔除;对全要素信息完整的数据行,抽取过程以记录的形式写入二维数组。

3. 新文件生成

将提取到的数据进行记录数统计,连同文件名检查/修正得到的数据文件信息一并写入新文件,存储到指定的数据准备区相应目录下。

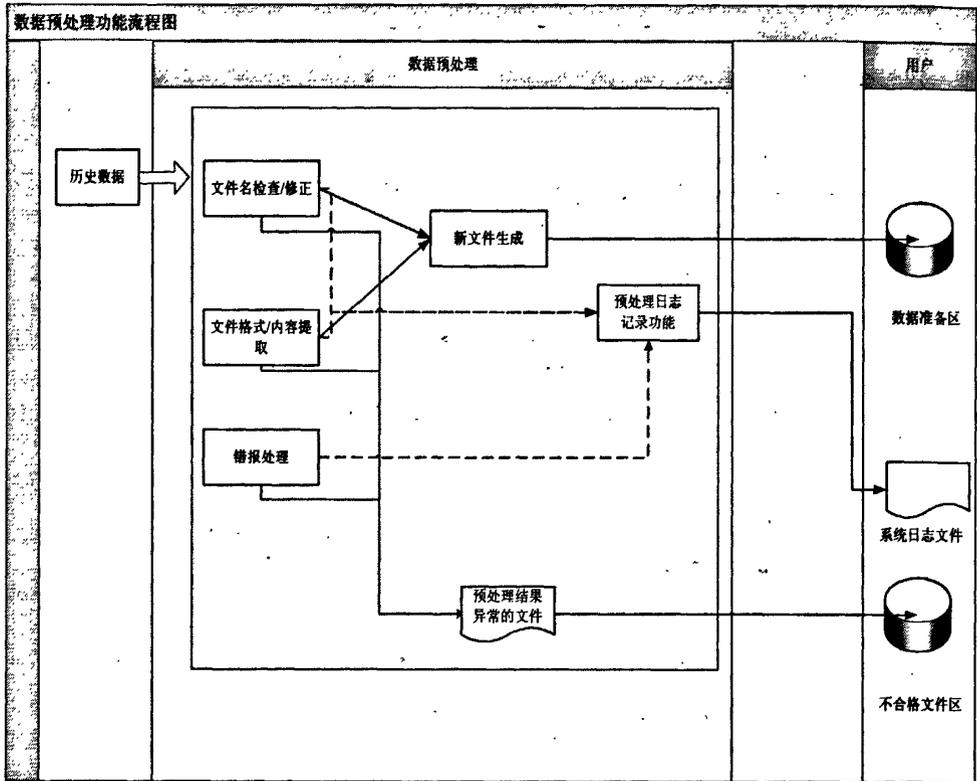


图 5-2 数据预处理功能流程

5.3 数据存储管理功能实现

与前文相同，本节以常规探空资料文件存储与数据入库为例，通过每一步子功能的实现过程，说明在系统开发过程中，是如何实现数据的存储管理的。

数据存储管理在流程上首先进行数据文件的获取与转存，然后分别进行元数据和文件要素数据(部分)的解析，最后进行数据保存。在代码编写过程中，全部的数据解析方法由 ParseMgr 类实现。对于元数据和文件要素数据的数据持久化，公司的架构采用了轻量级的 Hibernate 解决方案。

5.3.1 数据获取子功能的实现

数据获取子功能的主要任务是根据各类数据的获取策略监视数据准备区相应目录，当有新的数据到达时，将数据移入数据存储区。

具体的实现步骤如下：

1. 读取数据获取参数配置表，获取数据准备区的数据监视目录、时间间隔和、数据存储区的数据存储目录等参数信息；

2. 根据目录监视时间要求，周期性扫描数据准备区各目录是否有新的数据送达；
3. 若有新的数据送达，则将文件从数据准备区移到数据存储区指定的目录下；
4. 数据获取成功，则将数据获取状态信息写入数据存储流程状态表中，并将正常日志信息记入数据库日志表中；
5. 数据获取失败，则将错误日志信息写入数据库日志表中；
6. 每个目录数据获取完成后，则向数据存储调度管理子功能发出该目录数据获取完成的信息，由它进行下一步调度工作；
7. 一个目录数据获取完成后，进行下一个目录的数据扫描获取处理；
8. 若处理目录没有新数据送达，则直接进行下一个目录的数据扫描获取处理；

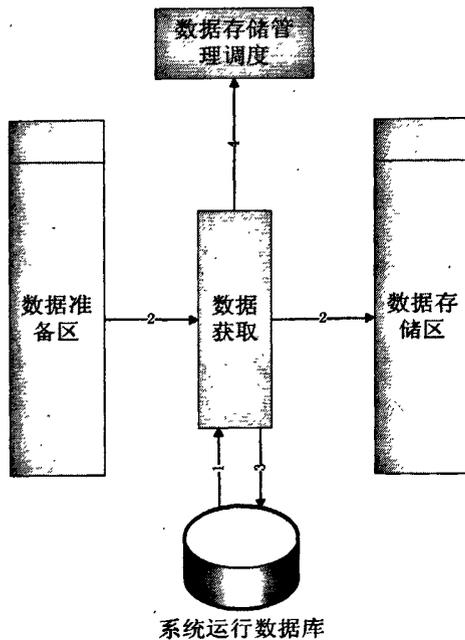


图 5-3 数据获取流程图

数据流程：

1. 数据获取参数信息；
2. 准备区新送达某数据；
3. 数据获取状态信息和日志信息入库；

4. 某目录数据获取完成信息;

5.3.2 元数据提取子功能的实现

元数据提取子功能实现的数据流程如图所示。

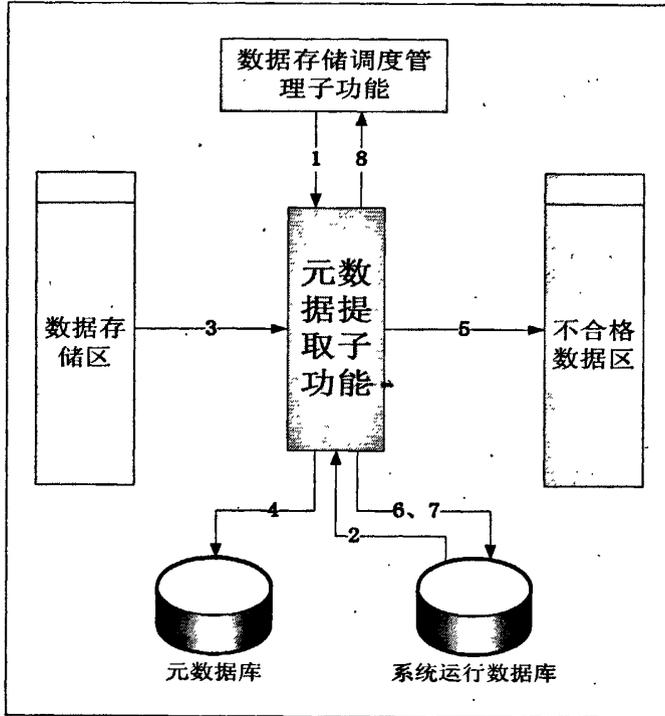


图 5-4 元数据提取数据流程图

数据流程:

1. 某目录数据获取完成信息;
2. 从数据存储流程状态表中获取此目录下未提取元数据的数据文件信息;
3. 依次从该存储目录读取需提取元数据的数据文件;
4. 将提取出的元数据 (包括数据文件关联信息) 写入元数据库;
5. 元数据提取失败的数据文件移到不合格数据区;
6. 将不合格数据信息写入不合格数据表;
7. 元数据提取状态信息和日志信息入库以及最新数据信息更新;
8. 上报存储调度子功能元数据提取完成状态;

```
//解析常规资料元数据伪代码
1. 建立metaRoutinedata对象,用以存储元数据
2. 建立essage对象,记录错误信息
3. 进行数据子类判断,符合要求的进行文件名解析,不符合要求文件,将错误计入eMessage对象返回;
4. 提取文件的文件名
    1) 判断文件名是否符合规范,不符合规范的文件转入不合格处理流程;
    2) 判断文件后缀名格式是否符合提取要求,对符合要求的文件进行提取操作,不符合标准时转入不合格处理流程;
    3) 规定格式化日期,严格控制输入,禁止不非法日期出现
        进行格式转换,并提取数据产生时间年份、月份、日、小时等值
5. 提取文件名
6. 提取文件大小的值
7. 将元数据信息写入metaRoutinedata对象
8. 将eMessage对象置0,表示没有错误发生
9. 返回;
```

5.3.3 数据解析入库子功能实现

结构化的数据文件需要将数据内容解析存储到数据库中,如常规资料、数值预报等。

要素数据解析入库子功能的实现流程如图所示。

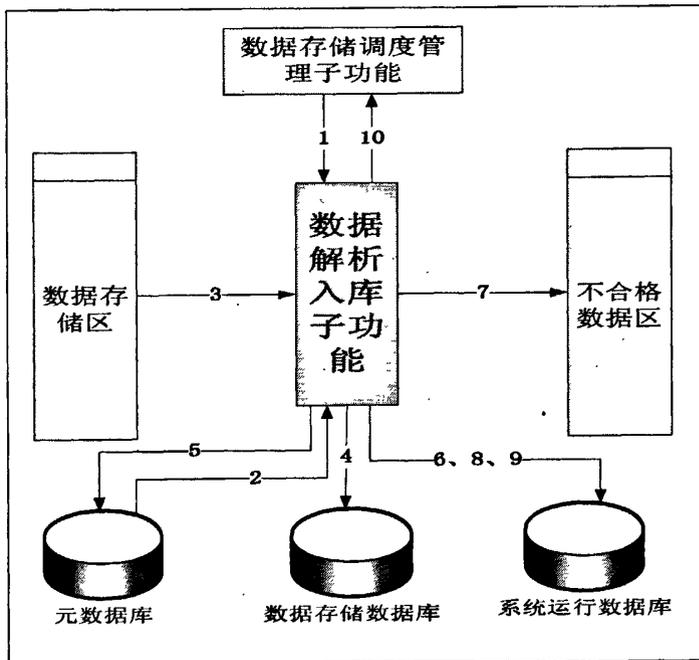


图 5-5 要素数据解析入库流程图

数据流程:

1. 某目录元数据提取完成信息;
2. 从元数据表中获取此目录下未入库的数据文件信息;
3. 依次从该存储目录读取需解析入库数据文件;
4. 将解析后的数据记录写入相关数据库;
5. 将数据入库是否成功状态写入元数据表;
6. 将数据入库是否成功状态写入数据存储状态信息表;
7. 若数据解析入库失败, 将文件移入不合格数据区;
8. 将不合格数据信息写入不合格数据表;
9. 日志信息入库;
10. 上报存储调度子功能解析入库完成状态;

// 解析探空资料要素解析算法伪代码

1. 建立metaRoutinedata对象, 调用meta元数据与文件内容比对
2. 建立eMessage对象, 记录错误信息
3. 建立list对象存储经过set对象处理的TRoutHigh对象
4. 建立head对象存储文件头数据
5. 判断文件是否符合要求(文件必须是存在的、非目录的), 对不符合要求情况记录错误并返回
6. 以lineno记录行数, 以定位文件可能出错地方
7. 以行为单位读取源数据文件, 记录文件头开始的行数
8. 以空格为分隔符解析文件头的元素
文件头要素数量不正确时报错, 进入不合格文件处理流程
9. 创建tempArray临时数组保存解析出来的元素
 - 1) 日期元数据与日期元素进行匹配, 符合要求的文件进行要素提取
规范日期对象格式
严格控制输入, 禁止非法日期出现
 - 2) 不匹配的文件填写记录错误进入不合格文件处理流程;
10. 开始处理文件体
 - 1) 建立stationnum变量保存区站号
 - 2) 对每个站的区站信息进行解析, 用临时变量存储区站号、经度、纬度、海拔高度、当前区站的层数; 区站要素数量不正确时记录错误及出错行数, 转入不合格文件处理流程

```

11. for循环处理区站各层
    {
        建立TRoutHigh对象, 用以存储部分资料要素
        建立TRoutHighId对象, 用以存储组成数据库主键的资料要素
        提取数据产生时间元素
        提取数据产生年份元素
        提取数据产生月份元素
        提取数据产生日元素
        提取数据产生时刻元素
        提取测站数元素
        提取区站号元素
        提取区站经度要素
        提取区站纬度要素
        提取区站海拔高度要素
        提取区站站点层数要素
        提取当前层数
        提取当前层气压
            提取当前层高度
            提取当前层温度
            提取当前层露点
            提取当前层风向
            提取当前层风速
            提取各信息的质量码
            提取文件名
            提取ID元素
            提取行号
        将提取结果写入TRoutHigh对象
        将TRoutHigh对象存入list对象
    }

12. 建立set对象, 唯一识别list对象中存储的TRoutHigh对象的TRoutHighId,
    以确保重复的区站及层信息只记录最新产生的数据
13. 将eMessage对象置0, 表示没有错误发生
14. 返回;
    
```

数据存储由类 DataCollectionMgr 实现。以下是 DataCollectionMgr 类中完整的探空数据元数据和要素数据的入库的实现代码

1. 元数据提取:

```

public ErrorMessage parseMetaRoutineData(TParaGetdata paraGetdata,
    String src, BaseMetaModel meta) {
    ErrorMessage message;
    File srcfile = new File(src);
    parseBefor(meta, paraGetdata);
    message = ParseMgr.parseMataRoutineData(srcfile, meta);
    parsemetaend(meta, paraGetdata, src);
    return message;
}

public ErrorMessage parseMeta(TaskObj task, ExecuteInfo executeInfo,
    int flag) {
    TParaGetdata paraGetdata = executeInfo.getParaGetdata();
    ErrorMessage message = new ErrorMessage();
    message.setFlag(true);
    String dataclass = StringUtil.nullToStr(paraGetdata.getDataclass());
    String datatype = StringUtil.nullToStr(paraGetdata.getDatatype());
    String datasubtype = StringUtil.nullToStr(paraGetdata.getDatasubtype());
    String src = task.getSrcPath();

    if (dataclass.equals("BASE") && datatype.equals("ROUT")) {
        TMetaRoutinedata metaRoutinedata = new TMetaRoutinedata();
        task.setMeta(metaRoutinedata);
        message = parseMataRoutineData(paraGetdata, src,
            metaRoutinedata);
    }

    return message;
}

```

2. 探空要素数据提取:

```

public ErrorMessage parseElementROUT_High(BaseMetaModel meta, List elements) {
    ErrorMessage message;
    String filepath = meta.getFilepath();
    File f = new File(filepath);
    message = ParseMgr.parseElementROUT_High(f, meta, elements);
    ParseEleEnd(elements);
    return message;
}

public ErrorMessage parseElements(TaskObj task, TParaGetdata paraGetdata) {
    task.setElements(new ArrayList());
    ErrorMessage message = new ErrorMessage();
    message.setFlag(true);
    String dataclass = StringUtil.nullToStr(paraGetdata.getDataclass());
    String datatype = StringUtil.nullToStr(paraGetdata.getDatatype());
    String datasubtype = StringUtil.nullToStr(paraGetdata.getDatatype());
    String dataattr = StringUtil.nullToStr(paraGetdata.getDataattribute());
    ....
    if (dataclass.equals("BASE") && datatype.equals("ROUT")
        && datasubtype.equals("HIGH") && dataattr.equals("GEN")) {
        message = parseElementROUT_High(task.getMeta(), task
            .getElements());
    }
    ...
    return message;
}

```

3. 存储元数据及要素数据:

```

//保存元数据
public ErrorMessage saveTaskMeta(BaseMetaModel meta) {
    String filepath = meta.getFilepath();
    ErrorMessage message = new ErrorMessage();
    try {
        getHibernateTemplate().saveOrUpdate(meta);
        message.setFlag(true);.....)
    }

//保存要素数据数据
private void saveEle(Session session, List elements) throws SQLException {
    Connection conn = session.connection();
    Statement stat = null;
    PreparedStatement pstat = null;
    try {
        String sql = getDelSql(elements);
        if (sql.equals("")) {
            return;
        }
        stat = conn.createStatement();
        stat.executeUpdate(sql);
        insertRoutHigh(elements, conn, pstat);
        ...}
}

```

4. 生成探空数据插入 SQL, 并执行入库操作:

```

public void getautoHighSql(TRoutHigh m, PreparedStatement stat)
    throws SQLException {
    if (m.getId().getFilename() == null) {
        stat.setNull(1, Types.VARCHAR);
    } else {
        stat.setObject(1, m.getId().getFilename());
    }
}

public void insertRoutHigh(List elements, Connection conn,
    PreparedStatement stat) throws SQLException {
    String sql = "insert into T_ROUT_HIGH (FILENAME, LINENO, PRODUCEDATE,
    PRODYEAR, PRODMONTH, " + " PRODDAY, PRODHOUR, STATIONCOUNT, STATIONNUM, TRANSTATIONNUM,
    STATIONTYPE, LONGITUDE, LATITUDE, " + " SEALEVELHEIGHT, TOTALLEVEL, LEVELNUM,
    PRESSURE, HEIGHT, TEMPERATURE, DEWPOINTTEMP, WIND, " + " WINDSPEED, PRESSUREQC,
    HEIGHTQC, TEMPERATUREQC, DEWPOINTTEMPOC, WINDQC, WINDSPEEDQC) values
    (? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ? , ?)";
    stat = conn.prepareStatement(sql);
    Iterator it = elements.iterator();
    while (it.hasNext()) {
        TRoutHigh m = (TRoutHigh) it.next();
        getautoHighSql(m, stat);
        stat.addBatch();
    }
    stat.executeBatch();
}

```

5. 数据存储管理功能的实现:

```

public boolean funcSave(TaskObj taskObj, TParaGetdata paraGetdata,
    TFlowArchive archive, TFlowDisqualification disqualification) {
    .....
    saveTaskMeta(...);
    saveEle(...);
    .....
}

public boolean funcparse(TaskObj taskObj, ExecuteInfo executeInfo,
    TFlowDisqualification disqualification, TFlowArchive archive) {
    .....
    parseMetaRoutineData(...);
    .....
}

public boolean funcEle(TaskObj taskObj, ExecuteInfo executeInfo,
    TFlowDisqualification disqualification, TFlowArchive archive) {
    .....
    parseElements(...);
    .....
}

public boolean processTask(TaskObj taskObj, ExecuteInfo executeInfo) {
    TParaGetdata paraGetdata = executeInfo.getParaGetdata();
    String putindb = StringUtil.nullToStr(paraGetdata.getPutindbflag());
    File srcFile = new File(taskObj.getSrcPath());

    TFlowDisqualification disqualification = new TFlowDisqualification(
        new TFlowDisqualificationId());

    initDisqualification(disqualification, paraGetdata, srcFile);
    TFlowArchive archive = new TFlowArchive();
    // 第一步 收集文件
    if (!funcCollection(paraGetdata, taskObj, disqualification, archive)) {
        return false;
    }

    // 第二步 元数据提取文件
    if (!funcparse(taskObj, executeInfo, disqualification, archive)) {
        return false;
    }

    // 第三步 解析要素数据
    if (putindb.equals("1")) {
        if (!funcEle(taskObj, paraGetdata, archive, disqualification)) {
            return false;
        }
    }

    // 第四步 保存元数据和要素数据
    if (!funcSave(taskObj, paraGetdata, archive, disqualification)) {
        return false;
    }

    return true;
}

```

实现探空数据的元数据与要素数据入库处理之后，在系统前台就可以检索/订
阅探空数据的数据资料了。图 5-6 至图 5-9 显示了检索结果。我们可以看到，元
数据与要素数据已经可以为在线研究人员提供服务了。

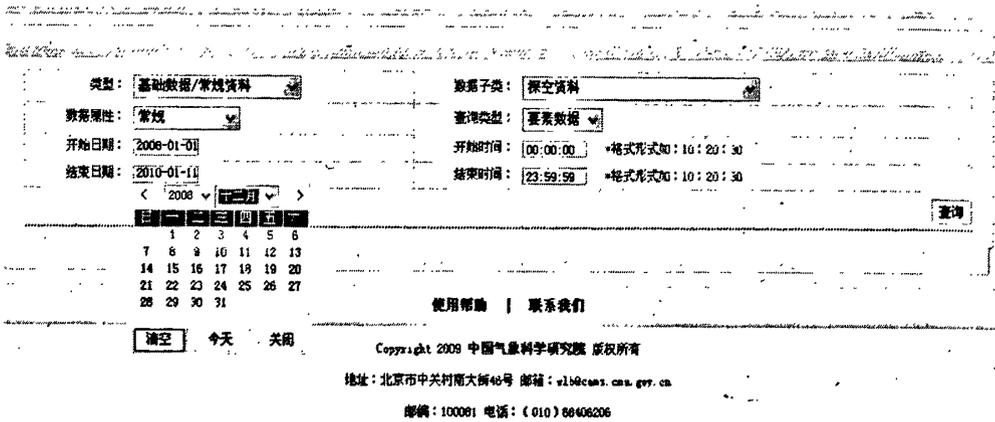


图 5-6 探空资料检索示意图

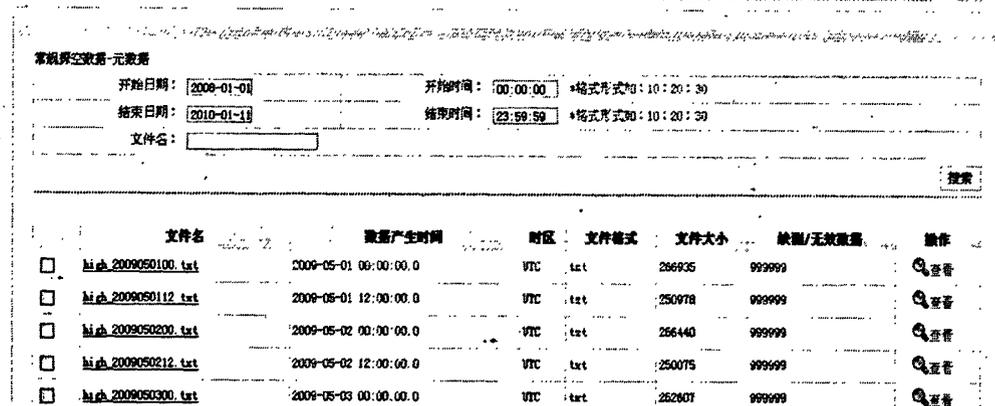


图 5-7 探空资料元数据检索结果

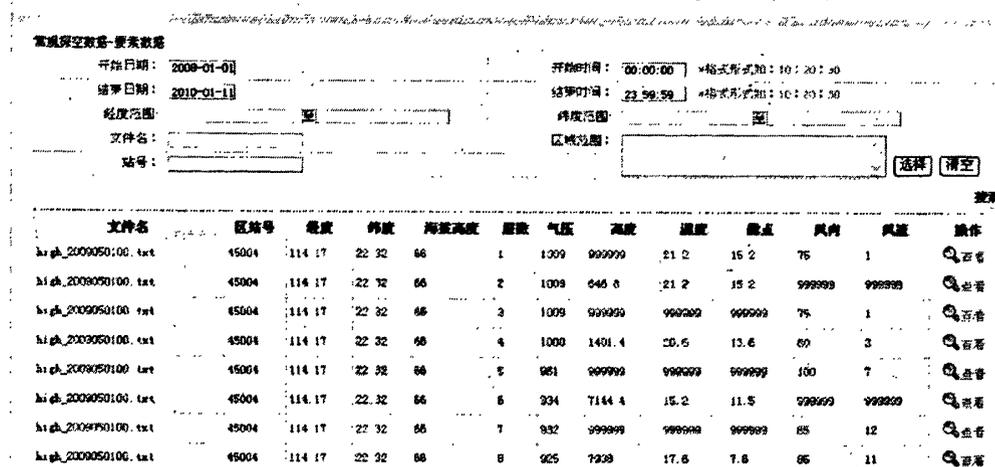


图 5-8 探空资料要素数据检索结果一

常规探空数据

文件名: b1gh_2009050100.txt
数据产生时间: 2009-05-01 00 00 00.0
测站号: 124
区站号: 45004
标准站号类别:
纬度: 22.32
经度: 114.17
海拔高度: 86
总层数: 109
层数: 1
高度: 99999
温度: 21.2
露点: 15.2
风向: 75
风速: 1
风向量码: 9
风速量码: 9
露点量码: 9
温度量码: 9

图 5-9 探空资料要素数据检索结果二

第6章 结论

灾害天气综合数据库系统在充分理解客户需求的基础上,目前仍处于研发之中,从初步的系统原型运行试验证明该系统成功的满足了气科院灾害天气数据存储服务需求和技术要求。但同时也应该注意到,由于代码和设计问题,某些地方出现比较严重的宕机现象。因此,充分利用现有硬件和软件资源,挖掘系统潜能,对系统进行进一步优化是下一步要解决的主要问题。解决这个问题需要从以下几个方面入手:首先,要规范设计和代码过程;其次,对影响系统性能的关键算法和代码进行优化;最后,研究系统与数据库的接口,充分利用商用数据库提供的优化方法。

参考文献

- [1] 灾害天气国家重点实验室简介
<http://www.lasw.cma.gov.cn/show.asp?id=1>
- [2] 中国气象科学研究院基本情况介绍 <http://www.cams.cma.gov.cn/>
- [3] 李集鸣 熊安元 气象科学数据共享系统研究综述 应用气象学报 2004 15:1-6
- [4] 王宏记 王海军等 省级气象信息综合数据库系统的设计与实现 暴雨灾害, 2008 27(3):283-284
- [5] 王京丽 谭晓东 张德政 大城市气象服务信息系统数据存储体系框架设计与实现 气象科技 2003 31(6):409
- [6] 高杰 胡欣等 MICAPS 3.0 用户使用手册 2008
- [7] 匿名 MICAPS 数据格式转换文档
- [8] 秦大河 孙鸿烈 中国气象发展战略研究总论卷 北京 气象出版社 2004
- [9] Standards Coordinating Committee of the Computer Society of the IEEE. IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology -Description. New York:The Institute of Electrical and Electronics Engineers, 1990
- [10] (美)Craig Larman 著 李洋 等译 UML 和模式应用 第3版 北京 机械工业出版社 2008
- [11] 史济民 顾春华 郑红 软件工程-原理 方法与应用 第3版 北京 高等教育出版社 2009
- [12] 系统项目组 灾害天气综合数据库系统建设项目需求分析说明书 2009
- [13] Geri Schneider, Jason P. Winters. Applying Use Case. Second Edition(影印版) 北京: 高等教育出版社 2003
- [14] 傅纯一 用例建模指南 IBM 中国有限公司软件部 2004
- [15] 王纪奎 成就存储专家之路-存储从入门到精通 北京 清华大学出版社 2009
- [16] Schmuck F, Haskin R. GPFS:A Shared-Disk File System for Large Computing Clusters. Proceedings of the FAST 2002 Conference on File and Storage

Technologies;2002:1-14

- [17] 总装备部 总装备部软件工程技术规范 北京 总装备部
- [18] 系统项目组 灾害天气综合数据库系统建设项目-综合数据库分册 2009
- [19] 系统项目组 灾害天气综合数据库系统建设数据存储架构设计书 2009
- [20] 系统项目组 灾害天气综合数据库系统建设项目数据分析文档 2009
- [21] Jonathan Lewis Cost-Based Oracle Fundamentals Apress 2006
- [22] 梁建全 周力 孟志勇 田利军 精通轻量级 JavaEE 框架整合方案 北京 人民邮电出版社 2008

致 谢

在本文即将完成之际，回想研究生阶段的学习生活，我衷心感谢我的导师张彩明教授。导师待人正直诚恳、关爱学生，在生活中他更像我们的父亲；在学术上，他深厚的理论素养、渊博的知识、严谨的治学态度，极大地感染了我们，我想这一切都会让我们感动铭记和受益终身的。在研究生阶段，老师对我的专业学习、课题研究、论文选题和撰写都给予了极大的关注和悉心指导，在此谨致以诚挚的敬意和衷心的感谢！

最后，感谢在我成长的过程中给予我帮助的所有老师、朋友和同学们。

学位论文评阅及答辩情况表

论文评阅人	姓名	专业技术职务	是否博导 (硕导)	所在单位	总体评价※	
		屠长河	教授	是	山东大学	A
		韩慧建	教授	是	山东经济学院	A
答辩委员会成员	姓名		专业技术职务	是否博导 (硕导)	所在单位	
	主席	韩作生	教授	是	山东经济学院	
	委员	张瑞华	副教授	是	山东大学	
		张立群	副教授	是	山东大学	
		赵台计	副教授	是	山东大学	
答辩委员会对论文的 总体评价※		良好	答辩秘书	王君	答辩日期	2010.5.30
备注						

※优秀为“A”；良好为“B”；合格为“C”；不合格为“D”。

