

密级:



中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

基于虚拟天文台的 HXMT 卫星数据检索发布 系统设计与实现

作者姓名: 姜旭

指导教师: 佟继周 副研究员

中国科学院空间科学与应用研究中心

学位类别: 工学硕士

学科专业: 计算机应用技术

研究所: 中国科学院空间科学与应用研究中心

2014 年 5 月

**Design and Implementation of HXMT Data Retrieval and
Publishing System Based on Virtual Observatory**

By

Jiang Xu

A Thesis Submitted to

University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the Degree of

Master of Computer Applied Technology

Center for Space Science and Applied Research

May, 2014

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，除了文中特别加以标注和致谢的地方外，论文中不包含任何其他他人已发表或撰写过的材料，也不包含为获得其它教育机构的别种学位或证书而大量使用过的材料。与我一同工作的人对本研究所做的任何贡献已在论文中作了明确的说明并表示谢意。

签名：_____ 日期：_____

关于论文使用授权的说明

本人完全了解培养单位有关保留、使用学位论文的规定，即：培养单位有权保留送交论文的复印件，允许论文被查阅和借阅；培养单位可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

签名：_____ 见证人：_____ 日期：_____

摘要

随着空间科学战略性先导科技专项的启动，我国预计发射六颗空间科学卫星，其中，硬 X 射线调制望远镜（HXMT）作为我国第一颗空间天文卫星，将填补我国高能天文观测在硬 X 射线波段的重大缺口。研制 HXMT 卫星数据检索发布系统，对于确保 HXMT 卫星数据的可访问性、提升天文数据资源的利用价值、最大化科研产出均具有重要意义。

目前，国际上 Chandra 卫星、Herschel 卫星、SDSS 巡天等诸多天文项目的数据检索发布系统均遵循虚拟天文台（VO）标准协议对外公开发布数据。天文数据资源经 VO 注册后，即可被用户通过 VAO 等门户网站访问获取到，从而实现其国际范围内的共享互访。因此，为了顺应天文学界数据检索发布系统的发展趋势，本文在充分调研 VO 发展现状、概念定义及标准协议的基础之上，结合 HXMT 卫星数据物理特性复杂、数据量大、衍生产品多等特点，设计并实现了基于 VO 技术框架的 HXMT 卫星数据检索发布系统。该系统既满足 HXMT 卫星数据发布需求，又将 HXMT 卫星数据融入 VO 环境，方便了国际间天文数据的共享共用。

首先，本文通过调研国际上天文数据检索发布系统的发展历程，得出引入 VO 技术的必要性，并对虚拟天文台的一系列标准协议进行重点梳理，包括 VO 体系结构、VOTable 数据格式和 VO 数据访问协议。

其次，本文结合遵循 VO 标准协议发布 HXMT 卫星数据的功能需求，兼顾考虑提供用户友好型界面、提升检索速率与响应速度的性能需求，设计了检索浏览、数据下载、可视交互三大核心功能模块。通过参考 IVOA 概念体系结构，构建了一套由资源层、服务层、用户层组成的体系架构。并在充分分析 HXMT 数据产品特点的基础之上，设计了数据实体文件、元数据信息、互访数据分离的分层数据模型。

随后，本文详细阐明了三大功能模块的实现方法和采用 HTM 索引优化数据库检索效率的具体步骤，展示了系统运行效果，并通过验证开发的锥形检索服务有效性来证明 HXMT 数据融入 VO 环境的可行性。

最后，本文总结了已完成的工作内容，并提出下一步的工作展望。

实践和应用结果表明，基于 VO 的 HXMT 卫星数据检索发布系统在解决天文数据资源互访、共享发布、检索浏览及异构应用集成方面均具有可操作性，对

我国空间天文卫星数据检索发布系统的研制具有参考意义。

关键词：虚拟天文台；HXMT 卫星数据检索发布系统；VO 标准协议；Spring MVC；Web 服务

Design and Implementation of HXMT Data Retrieval and Publishing System Based on Virtual Observatory

Abstract

With the start of the strategic leading science and technology project in space science, six space science satellites are expected to launch in our country. As China's first space astronomy satellite, Hard X-ray Modulation Telescope (HXMT) will fill the significant gap of high-energy astronomical observations in hard X-ray band. Meanwhile, the development of HXMT data retrieval and publishing system is of great benefit to ensure HXMT data's accessibility, enhance its utilization value and maximize the scientific research output.

Currently, many astronomical projects have complied with the Virtual Observatory (VO) standard protocols to develop the data retrieval and publishing system to release data publicly, such as Chandra X-ray Observatory, Herschel Space Observatory, Sloan Digital Sky Survey and so on. Once registered in VO registration system, the astronomical data of various projects can be obtained by the VO portals quickly and easily, realizing the data sharing within the international scope. Therefore, this paper adopted VO technologies into HXMT data retrieval and publishing system design and implementation, which both met the project requirements of publishing HXMT satellite data, and integrated HXMT data into VO environment for international astronomical data sharing.

Firstly, this paper did sufficient research on the domestic and overseas development of astronomical data retrieval and publishing system. Meanwhile, it expounded the VO standard protocols in detail, including VO architecture, VOTable data formats and VO data access protocols.

Secondly, combining the requirement of introducing VO technology to publish HXMT data, and providing a user-friendly system with high retrieval rate and response speed, this paper confirmed three core function modules, including retrieval browsing module, data downloading module and interactive visualization module.

Referring to IVOA architecture, it constructed the HXMT system architecture with the resource layer, service layer and user layer. Furthermore, it designed a hierarchical data model with the separation of HXMT data entity file, metadata information and interoperate data.

Thirdly, the paper illuminated both the implementation of three function modules and the steps of using HTM index to optimize the database retrieval efficiency. Then, it displayed the system operating interface, and demonstrated the feasibility of integrating HXMT data into VO environment by the validity of the Cone Search service.

Finally, the paper summarized the work which has been completed and proposed the future work aspect.

Experiment results showed that the HXMT data retrieval and publishing system is operational in data interoperability, sharing, retrieval and accessing, which is of great significance in the development of other space astronomical satellite data retrieval and publishing systems in China.

Keywords: Virtual Observatory; HXMT data retrieval and publishing system; VO standard protocol; Spring MVC; Web Service

目录

摘要	I
ABSTRACT	III
目录	V
图索引	VII
表索引	IX
第一章 绪论.....	1
1.1 课题背景及意义	1
1.2 研究现状	3
1.2.1 天文数据检索发布系统的发展历程.....	3
1.2.2 虚拟天文台的发展现状.....	6
1.3 研究内容	13
1.4 本文组织结构	14
第二章 虚拟天文台基础理论与应用技术研究	15
2.1 虚拟天文台标准与协议	15
2.1.1 体系结构.....	15
2.1.2 VOTable 数据格式.....	19
2.1.3 数据访问协议.....	21
2.2 应用技术	23
2.2.1 SOA 架构.....	23
2.2.2 MVC 模式.....	25
2.2.3 Spring 框架.....	26
2.3 本章小结	27
第三章 系统设计	29
3.1 系统设计目标	29
3.2 系统功能设计	29

3.3	体系架构设计	30
3.4	数据模型设计	32
3.4.1	数据产品分析.....	32
3.4.2	分层数据模型设计.....	40
3.4.3	数据库设计.....	42
3.4.4	数据库索引设计.....	45
3.5	本章小结	49
第四章	系统实现与运行.....	51
4.1	技术开发框架	51
4.1.1	开发环境.....	51
4.1.2	技术架构.....	51
4.2	主要功能实现	54
4.2.1	检索浏览.....	54
4.2.2	数据下载.....	62
4.2.3	可视交互.....	64
4.2.4	数据库实现.....	66
4.2.5	数据库索引实现.....	66
4.3	系统运行	70
4.3.1	运行环境.....	70
4.3.2	运行界面.....	70
4.4	本章小结	73
第五章	总结与展望	75
5.1	工作总结	75
5.2	创新点	76
5.3	工作展望	77
参考文献.....		79
发表文章目录.....		83
致谢		85

图索引

图 1.1 CHANDRA 卫星数据 FTP 目录	4
图 1.2 SKYSERVER 图片检索界面	5
图 1.3 VAO 检索界面	6
图 1.4 IVOA 成员图	10
图 1.5 VODESKTOP 应用软件	12
图 1.6 SKYMOUSE 检索界面	13
图 2.1 0 级 VO 概念体系结构	15
图 2.2 1 级 VO 概念体系结构	15
图 2.3 2 级 VO 概念体系结构	16
图 2.4 ASTROGRID 组件	17
图 2.5 CHINA-VO 体系结构	18
图 2.6 VOTABLE 数据模型	19
图 2.7 VOTABLE 实例	20
图 2.8 VOTABLE 解析包	20
图 2.9 锥形检索概念图	21
图 2.10 WEB 服务协议栈	24
图 2.11 WEB 服务调用流程	24
图 2.12 SPRING 框架	26
图 3.1 HXMT 卫星数据检索发布系统的体系架构	30
图 3.2 顶级目录结构	33
图 3.3 HXMT 1L 级数据目录结构	34
图 3.4 HXMT 1S 级数据目录结构	35
图 3.5 HXMT 2L 级数据产品目录结构	36
图 3.6 VOTABLE 设计	42
图 3.7 E-R 图	43
图 3.8 数据库物理设计模型图	45
图 3.9 HTM 天区划分方法	48
图 3.10 HTM 不同划分级次对应的单元数和单元大小	48

图 3.11(A) HTM 编码方法	图 3.11 (B) HTM 编码方法.....	48
图 4.1 系统技术架构		53
图 4.2 系统类图		53
图 4.3 检索浏览功能模块流程图		55
图 4.4 WEBSERVICE 实现类		57
图 4.5 SEARCH 实现类		58
图 4.6 WEB.XML 文件		58
图 4.7 CXF-BEANS.XML 文件		59
图 4.8 CONESEARCH SCHEMA 文件		59
图 4.9 WSDL 文件		60
图 4.10 HXMT 卫星数据融入 VO 应用环境示意图		62
图 4.11 数据下载模块流程图		63
图 4.12 可视交互模块流程图		65
图 4.13 数据库 SQL 脚本		66
图 4.14 采用 HTM 索引方法的锥形检索流程图		67
图 4.15 HXMT 检索浏览界面		70
图 4.16 符合 SCS 协议的 VOTABLE 格式锥形检索结果数据		71
图 4.17 HXMT 数据下载界面		71
图 4.18 ALADIN 可视检索界面		72
图 4.19 ALADIN 点源标注界面		72
图 4.20 ALADIN 可视浏览界面		73

表索引

表 3.1 HXMT 科学数据产品级别	33
表 3.2 HXMT 卫星和 CHANDRA 卫星数据分级对比	36
表 3.3 HXMT 卫星数据产品文件类型列表	38
表 3.4 分层数据模型	40
表 3.5 HXMT 卫星 1L 级数据产品实体文件信息表	43
表 3.6 HXMT 卫星 1L 级产品元数据表	44
表 3.7 HXMT 卫星 1L 级数据产品组织信息表	44
表 3.8 用户信息表	45
表 4.1 实验检索条件	69
表 4.2 HTM 索引效率测试结果对比	69

第一章 绪论

1.1 课题背景及意义

众所周知,由于受地球大气的影响,人们通过地面望远镜只能观测到有限的光学和射电波段。空间天文卫星的出现,使人们摆脱了地球大气的束缚,将观测范围扩展到了 X 射线、 γ 射线、紫外和红外波段,从而促使空间天文学兴起,并逐步开启了多波段大尺度观测、数据资源繁杂丰富的空间天文新时代。

为了紧跟时代发展步伐,缩小与发达国家在空间天文学方面的差距,我国启动了“空间科学战略性先导科技专项”(简称“空间科学先导专项”),计划在十二·五期间发射六颗空间科学卫星。其中,硬 X 射线调制望远镜(Hard X-ray Modulation Telescope,简称 HXMT)卫星、中法合作天基多波段天文变源监视器(Space-based Multi-band Astronomical Variable Object Monitor,简称 SVOM)卫星、暗物质粒子探测卫星均为空间天文卫星。

HXMT 卫星预计于 2015 年发射,将成为我国第一颗自主研发的空间天文卫星^[1]。其科学目标是实现高灵敏度宽波段 X 射线巡天,研究致密天体和黑洞强引力场中动力学和高能辐射过程,研究 X 射线脉冲星自主导航可行性^[2]。HXMT 卫星将搭载高能 X 射线望远镜(HE)、中能 X 射线望远镜(ME)、低能 X 射线望远镜(LE)和空间环境监测器(SEM)四种有效载荷。预计在 4 到 6 年的寿命里,通过巡天观测、小天区观测和定点观测三种工作模式产生共计约 22TB 的原始观测数据。遵循国际天文数据分级标准规范,对用户发布的 HXMT 卫星数据划分为 1 级和 2 级科学数据产品。1 级科学数据产品包含 1 级提案观测数据(1L)和 1 级巡天观测数据(1S),2 级科学数据产品包括光变曲线、能谱等具有科学意义的数据产品。

目前,国际上诸多空间天文卫星项目,例如美国的 Hubble 空间望远镜、Chandra X 射线天文台、Fermi 伽玛射线天文台、Swift 伽玛射线暴卫星,欧洲的 Herschel 红外天文台、Planck 宇宙微波背景天文卫星、XMM-Newton X 射线天文台、Integral 伽玛射线天文台等,均遵循虚拟天文台(Virtual Observatory,简称 VO)标准协议开发各自的数据检索发布系统。通过虚拟天文台可将全球范围内

的天文研究资源进行无缝透明连结^[3]，实现不同空间天文项目数据资源之间的互访问、互操作，从而提升数据利用价值，最大化科研产出。因此，本文也引入 VO 技术进行 HXMT 卫星数据检索发布系统的研制与开发。

设计并研发基于 VO 的 HXMT 卫星数据检索发布系统，主要具有如下意义：

1、开展 HXMT 卫星数据组织管理研究，为其他空间天文卫星数据的归档管理提供参考价值。

针对 HXMT 卫星数据物理特性复杂、数据量大、衍生产品多等特点，开展 HXMT 卫星数据组织管理，包括进行数据分级定义，设计归档目录结构，确定数据存储格式，梳理数据产品种类等，从而对今后我国其他空间天文卫星的数据归档管理具有参考意义。

2、设计并实现 HXMT 卫星数据检索发布系统，为后续研制其他空间天文卫星数据检索发布系统提供技术支撑。

研制的 HXMT 卫星数据检索发布系统是我国“空间科学先导专项”地面应用支撑系统的一个重要子系统，可将数据种类繁多的 HXMT 卫星数据发布出去，支持用户对 HXMT 各级数据产品的检索访问、浏览下载与可视交互。而国内对空间天文卫星数据检索发布系统的研制几乎为空白，因此，设计并实现用户友好型的 HXMT 卫星数据检索发布系统可以为我国其他空间天文卫星数据检索发布系统的研制提供技术支持。

3、遵循 VO 标准协议进行系统研制，以实现国际间空间天文数据共享、提升数据利用率。

基于 VO 技术框架进行系统研制，对 IVOA 概念体系架构进行适当裁剪后构建原型系统的体系架构，采用 VOTable 格式数据与 Aladin 开源可视化软件进行可视交互，开发符合 SCS (Simple Cone Search) 协议的锥形检索服务提供数据访问接口。从而将 HXMT 数据融入 VO 环境，实现与不同天文项目数据资源之间的共享互访。这对提升数据利用率、最大化科研产出具有重要意义。

1.2 研究现状

1.2.1 天文数据检索发布系统的发展历程

从传统的采用纸张记录天文数据,到现在采用计算机技术进行数字化归档发布管理,经历了一个漫长的过程。在不同历史时期,各个国家的天文项目会采用不同的方式来设计并研制天文数据检索发布系统,提供数据下载、检索浏览等功能,实现对外公开发布数据。按时间顺序和应用技术进行划分,天文数据检索发布系统主要经历了 FTP 方式、关系型数据库和虚拟天文台技术三个阶段。

1、FTP 方式

70 年代开始,微型计算机开始由军事应用转到商业应用层面上,一些天文学家开始初步尝试利用计算机存储和管理天文数据,于是出现了基于文件系统的天文数据归档发布方式。80 年代起,随着 Internet 网络的普及,各国的天文项目开始通过 FTP (File Transfer Protocol) 方式来发布文件系统中归档存储的天文数据。FTP 即文件传输协议,是 TCP/IP 协议族中的协议之一,能提供文件共享功能,支持远程数据访问,并能可靠高效地传输数据。基于 FTP 协议,用户可以方便地上传、下载数据文件。

目前,空间天文领域已很少采用 FTP 方式发布数据,但仍有部分天文项目沿用 FTP 方式。例如,美国的 Chandra 卫星项目至今仍保留着 FTP 形式的数据检索发布系统,如图 1.1 所示(参考网址 <ftp://cdaftp.harvard.edu/pub/>)。采用 FTP 方式发布数据后,用户可通过 FTP 地址(ftp://用户名:密码@FTP 服务器 IP 或域名:FTP 命令端口/路径/文件名)直接访问到底层数据存储系统,层层目录点击进去后即可找到所需数据,检索方式比较单一简洁。但因用户只能通过 FTP 地址进入文件系统进行查找,缺乏相应的检索机制,而且必须对数据归档目录结构有充分的了解,才能定位到所需数据,所以导致检索效率低下、用户友好性较差。

/pub/ 的索引

名称	大小	修改日期
 [上级目录]		
 OOREADME	5.6 kB	10-2-1 上午12:00:00
 byobsid/		10-2-1 上午12:00:00
 cal/		13-2-11 上午12:00:00
 er/		13-1-16 上午12:00:00
 science/		09-8-5 下午5:32:00
 stage/		09-8-28 上午4:00:00

图 1.1 Chandra 卫星数据 FTP 目录

2、引入关系型数据库

90 年代以后，关系型数据库技术逐步发展起来，并出现了许多成熟的商业数据库产品，如 DB2、Oracle、SQL Server、MySQL、PostgreSQL 等。随着关系型数据库的广泛应用与普及，各天文项目开始基于关系型数据库开发天文数据检索发布系统，例如 NED（NASA/IPAC Extragalactic Database）、MAST（Mikulski Archive for Space Telescopes）、SDSS（Sloan Digital Sky Survey）、ADS（SAO/NASA Astrophysics Data System）等项目均基于关系型数据库研制了大批极具影响力的数据检索发布系统。后台负责抽取文件系统中天文数据的元数据信息和部分文件信息并将其导入到关系型数据库中，前端则通过 Web 界面或客户端软件将数据库中的数据对外发布出去，并与用户直接交互，从而方便用户检索、定位、访问数据，提升数据检索效率。

以著名的 SDSS 项目为例，SDSS 项目组采用 SQL Server 数据库存储海量的斯隆数字巡天观测数据，并针对 SDSS 数据自身特性研制了一套深受好评的 SDSS 数据检索发布系统——SkyServer，如图 1.2 所示。SkyServer 系统具有较为多元的检索查询方式，包括锥形检索、组合条件检索和 SQL 语句检索等多种方式以供用户选择；同时，提供友好的可视交互界面，例如，用户在星空图上点击某一坐标位置，可以显示对应的快视图片以及详细的星表数据信息。迄今为止，SDSS 项目已经稳定运行十多年，共计释放发布了 10 批次的数据，今后也将陆续发布最新的巡天观测数据。SDSS 巡天数据成为国际天文学界广为研究与使用的数据集之一，同时，SDSS 项目中应用的数据检索发布模式也为后续许多天文项目所借鉴。

目前，大部分空间天文项目均支持采用关系型数据库来存储底层数据产品信

息。尽管此类系统较之前的 FTP 方式已经有了很大的改善,但仍存在一大弊端,即各个国家的不同天文项目按照自身项目需求及规定指标开发自成一体的数据检索发布系统,不论是底层数据存储格式和归档目录结构,还是前端检索条件设定、后台数据检索处理方法等,均存在不一致。这种差异性使得不同项目之间天文数据共享层次较低,数据利用率低下,容易造成科研资源的极大浪费。



图 1.2 SkyServer 图片检索界面

3、虚拟天文台技术

为了提升全球范围内空间天文数据的共享利用率,有必要对不同项目中天文数据存储、发布、访问模式进行标准化管理。21 世纪初,一个跨天文学、信息学、计算机学科的概念——虚拟天文台被各国天文学家提出。同时,成立了国际性权威组织机构——国际虚拟天文台联盟 (International Virtual Observatory Alliance, 简称 IVOA), 负责为国际天文界制订统一的天文数据互访问互操作标准协议,并推动这一系列标准协议在实际项目中的应用与实施,使得对天文数据的收集、归档、存储、处理、发现、获取这整个数据生命周期流程都能在标准的 VO 框架下进行。

当前,各国天文项目纷纷遵循 VO 标准协议进行天文数据检索发布系统的研发,提供符合 VO 标准接口的数据源。例如美国的 HEASARC (High Energy Astrophysics Science Archive Research Center) 中心、IRSA (NASA/IPAC Infrared

Science Archive) 中心, 美国的 Chandra 卫星、Hubble 望远镜、SDSS 巡天项目、欧洲的 Herschel 卫星、XMM-Newton 卫星等上百家天文数据中心或空间天文项目组均声明支持 VO 标准协议。各天文项目的数据资源和检索浏览服务经 VO 注册后, 可被 VO 体系内的 VAO (Virtual Astronomical Observatory, <http://www.usvao.org/>, 如图 1.3 所示) 等门户系统方便快捷地检索浏览到, 从而屏蔽了不同数据检索发布系统之间的一致性, 实现了分布式异构数据集和各类服务之间的互操作、互访问, 提升国际间天文数据的共享利用率。

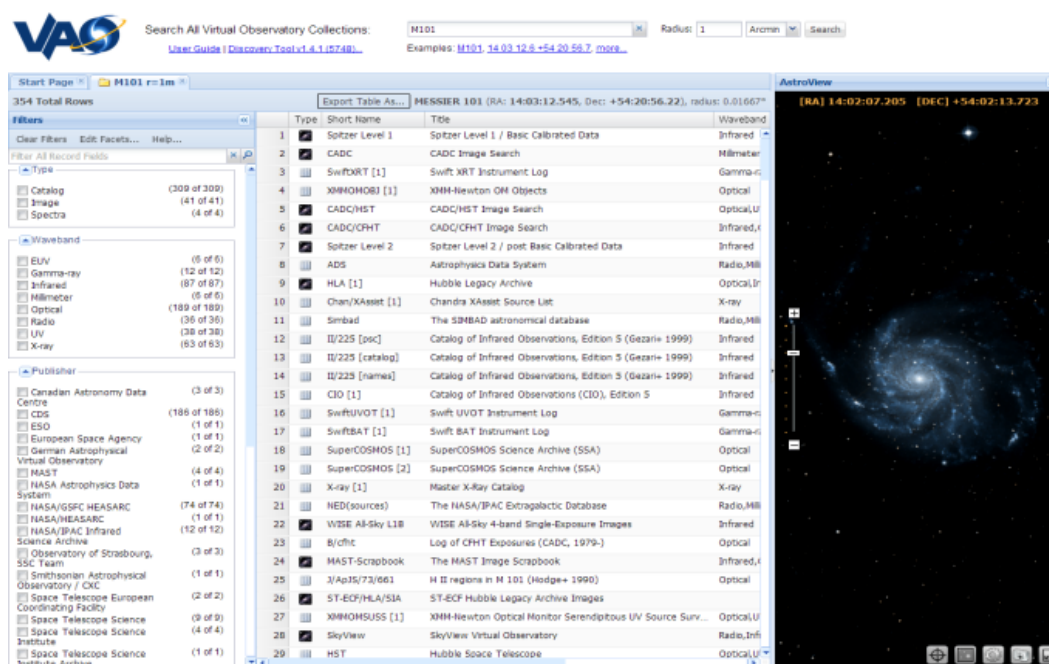


图 1.3 VAO 检索界面

综上所述, HXMT 卫星数据检索发布系统采用关系型数据库来存储元数据信息, 并在研制过程中引入 VO 技术, 是顺应天文学界数据检索发布系统发展趋势的表现, 具有一定的必要性。下面着重对虚拟天文台的发展情况进行介绍。

1.2.2 虚拟天文台的发展现状

1、虚拟天文台的产生背景

随着国际上各个波段巡天观测、定点观测等大规模天文项目的推进, 人造卫星、天文望远镜等探测手段的不断发展, 可访问的天文观测数据的数据量呈不断上升趋势, 使天文科学研究进入“数据雪崩”时代, 天文学也逐渐发展为典型的数据密集型学科^[2]。信息时代的天文数据本身具有如下特点:

（1）大数据量

如同在计算机领域中，计算机性能会随时间指数增长的摩尔定律一样，飞速发展的信息技术使得天文探测数据量的增长速度也遵循着摩尔定律^[4]，呈现指数增长趋势。例如，我国独立自主研制成功的LAMOST望远镜（Large Sky Area Multi-Object Fiber Spectroscopy Telescope，大天区面积多目标光纤光谱望远镜）每晚获得的原始观测数据大约为20GB，按五年的生命周期来算整个项目的原始数据将达到30TB，再附加上处理得到的一维光谱数据、星表数据等数据产品，总数据量将达60TB^[5]。启动的“空间科学战略性先导科技专项”预计每天都将产生GB甚至TB级别的数据。美国计划在十年发展规划中优先推荐建造的LSST望远镜（Large Synoptic Survey Telescope，大口径综合巡天望远镜），预计每15秒获得一幅6GB的天区图，按每晚产生2000幅图像来算，每晚生成的数据量将达到30TB，持续工作时间初步预计为10年，那么，LSST整个项目的图像数据存储总量将达100~200PB，星表存储总量达20~40PB^[6]。此外，TMT（Thirty Meter Telescope）、SKA（Square Kilometre Array）等正在由多国合作建设的天文项目也预计具有十分庞大的数据产出量。

天文数据呈海量增长的直接原因主要包含如下三个方面：世界各国各类天文观测项目的实施启动，大批天文探测设备的投入运行，都直接导致海量天文数据的产生；对天文原始探测数据进行分析处理后，生成了种类繁多的各级数据产品；天文数值模拟与仿真实验也催生了大量模拟数据^[7]。

（2）高复杂度

天文数据种类和数据处理方式的复杂度随天文数据量的增长而增加。天文数据类型主要包括：记录天体位置和参数等信息的星表数据、反映目标天体物理信息的光谱数据、直接拍摄天空某一区域的图像数据、时域天文观测生成的时序数据、进行计算机仿真模拟实验所产生的数值模拟数据，以及各类软件工具、科学文献等^[8]。不同类型数据相应的处理方式也依天文观测项目的不同而各有差异。

（3）全波段

天文数据涵盖 γ 射线、X射线、紫外、光学、红外、射电波段等全波段数据^[9]。若提供全波段数据融合环境，在更广泛的时空范围内进行天文研究，想必天

文数据利用率、科研产出量都将持续增加。

天文数据的发布特性具有以下两点：

（1）开放性

国际上各大天文项目基本执行相似的数据政策，即在数据发布过程中可划分为私有期和公有期两个阶段。当观测数据产生后，会在固定年限的私有期内仅提供给局部范围内的科研人员使用，确保数据提供者拥有充裕的时间处理分析数据，得到相应的科研回报。私有期过后，便可遵循一定的规则面向外界公开发布数据，从而使得世界范围内的科研人员或天文爱好者均有机会共享和使用这些宝贵的天文数据资料。私有期时间通常为一年，但也会根据具体项目实施情况、用户约定权限等条件作相应调整。总之，国际上许多大型天文项目的观测数据、辅助数据等天文数据资源均能及时在互联网上发布，这是其他学科数据所无法比拟的特点^[10]，同时，天文数据的开放性为数据共享提供了良好的基础。

（2）合理归档

在数据归档存储方面，为了统一数据格式，国际天文学联合会（International Astronomical Union，简称 IAU）制定了 FITS（Flexible Image Transport System）标准。目前，绝大多数天文数据均会保存为 FITS 格式。各国天文数据中心在数据的归档、发布方面做已做了大量的工作^[11]，使得大部分天文数据都拥有良好的归档组织形式。但由于缺乏统一的数据交换、发布标准，导致不同天文项目的数据检索发布系统之间无法互通互联，难以实现数据共享与融合。

基于上述天文数据的自身特点和发布特性可知，天文数据的海量产出和高复杂度对计算机存储能力、数据处理效率、IO 带宽通信能力等多方面提出了挑战，同时，天文数据的全波段特性以及在发布过程中的开放性和良好归档特征为实现大规模分布式天文数据的共享与应用奠定了一定基础，也提出了新需求。空间天文领域已经进入一个数据爆炸的时代，面对这些多波段海量数据，天文学家如何对其进行检索浏览和处理分析，各数据中心如何协调好各方面资源，对数据进行归档存储和共享发布，这些已成为各国科研人员共同面对的问题。

在此背景下，美国的天文科研人员于 1999 年首先提出虚拟天文台的概念，并在十年科学发展规划中将建立国家虚拟天文台作为最优先发展的科研项目^[12]。

此后，不同国家不同机构陆续提出愈来愈多的虚拟天文台计划。为了规范化管理虚拟天文台技术，避免天文学者进行重复性工作，于是在 2002 年 6 月德国召开的“走向国际虚拟天文台”国际会议上，成立了 IVOA 联盟（International Virtual Observatory Alliance）。目前在 IVOA 主导下已经制定并推广了数十项 VO 标准与协议。虚拟天文台的提出推进了天文观测、地面测控、数据应用、先进技术的深度一体化，有利于实现天文数据应用价值的最大化。

2、虚拟天文台的概念定义

什么是虚拟天文台？虚拟天文台通常定义为符合 VO 标准协议的系统所组成的数据密集型网络化天文研究环境，将各种天文研究资源无缝透明连结起来^[13]，使得科研人员足不出户，通过 VO 门户网站即可访问、下载到不同天文项目的数据资源，实现一站式访问。此外，在充分调研虚拟天文台基础理论与技术框架的基础之上，本文提出虚拟天文台还有两方面的含义。

其一是指虚拟天文台标准与协议，包括数据访问协议、数据格式、数据模型等。VO 标准数据访问协议包括简单锥形检索（SimpleConeSearch, SCS），简单图像访问（Simple Image Access, SIA）、简单光谱访问（Simple Spectral Access, SSA）、表访问协议（Table Access Protocol, TAP）等，数据格式为 VOTable 格式，数据模型包括 IVOA 光谱数据模型、观测数据模型、仿真数据模型、时空坐标系元数据模型等^[14]。为了有效融入 VO 环境，各国天文项目在数据检索发布系统研制过程中需要遵循这一系列的 VO 标准协议。

其二是指负责制定 VO 标准协议的国际权威组织机构——IVOA。各个国家可以申请加入 IVOA 组织并承诺遵循规则发布天文数据。截至 2013 年，IVOA 成员已经增至 19 个，如图 1.4 所示^[15]。至今为止，IVOA 每年组织两次国际研讨会，通过不同参与国天文科研人员之间的探讨交流来制定 VO 协议标准并推动其实施，同时，进一步消除分歧，解决了许多互操作问题。

项目	地区	网址
阿根廷虚拟天文台(NOVA)	阿根廷	http://nova.conicet.gov.ar/
亚美尼亚虚拟天文台(ArVO)	亚美尼亚	http://arvo.sci.am
英国虚拟天文台(AstroGrid)	英国	http://www.astrogrid.org/
澳大利亚虚拟天文台(Aus-VO)	澳大利亚	http://aus-vo.org.au/
巴西虚拟天文台(BRAVO)	巴西	http://www.lna.br/bravo/
中国虚拟天文台(China-VO)	中国	http://www.china-vo.org
加拿大虚拟天文台(CVO)	加拿大	http://www.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/cvo/
欧洲空间局(ESA-VO)	欧空局	http://esavo.esac.esa.int/
欧洲虚拟天文台(EURO-VO)	欧盟	http://www.euro-vo.org/
德国物理虚拟天文台(GAVO)	德国	http://www.g-vo.org/
匈牙利虚拟天文台(HVO)	匈牙利	http://hvo.elte.hu/en/
日本虚拟天文台(JVO)	日本	http://jvo.nao.ac.jp/
美国虚拟天文台(VAO)	美国	http://www.usvao.org
法国虚拟天文台(France-VO)	法国	http://www.france-vo.org/
俄罗斯虚拟天文台(RVO)	俄罗斯	http://www.inasan.rssi.ru/eng/rvo/
西班牙虚拟天文台(SVO)	西班牙	http://svo.cab.inta-csic.es/
乌克兰虚拟天文台(UKRVO)	乌克兰	http://www.ukr-vo.org/
意大利虚拟天文台(VOBS.IT)	意大利	http://vobs.astro.it
印度虚拟天文台(VO-I)	印度	http://vo.iucaa.ernet.in/%7Evoi/

图 1.4 IVOA 成员图

3、虚拟天文台国内外发展情况

虚拟天文台主要朝着两个方向发展，一是完善虚拟天文台体系架构，制定互操作标准与协议，从而使得依据 VO 协议族研制的系统软件或应用服务之间能够实现互访问与互操作；二是利用先进的计算机技术，开发天文领域用于学科研究分析的特色软件工具，如 Aladin、Topcat 可视化软件等。这两个方向是相辅相成、相互促进发展的。

不同国家的虚拟天文台会依据自身科研实力及经费条件来确定其发展路线，下面将针对几个国家的 VO 项目发展情况逐一进行介绍。

(1) 美国虚拟天文台

美国虚拟天文台 (US Virtual Astronomical Observatory, 简称 VAO) 是由 NSF (National Science Foundation, 美国国家科学基金) 和 NASA (National Aeronautics and Space Administration, 美国国家航空航天局) 资助，采用计算机、数据库及网络信息技术构建的虚拟天文台。VAO 汇聚了众多知名天文项目归档系统中的数据集，如 Chandra 卫星、Hubble 望远镜、SDSS 巡天项目、2MASS 巡天项目、CDS Vizier、NASA HEASARC 中心跨波段数据归档系统等。这些分布在不同地点、由不同天文项目获取的数据集，通过 VAO 融合在一起，覆盖了整个天区，

跨越了多个波段。

目前, VAO 的主要科学和技术成就反映在标准协议制定和软件应用开发两方面。在标准协议制定方面, VAO 提出了锥形检索 (SCS)、简单图像访问协议 (SIA)、VOTable 数据格式、统一内容描述标准 (Unified Content Descriptor, UCD) 等核心标准。在软件应用开发方面, VAO 硕果累累, 相继研发出如下学科应用工具: ①数据检索工具: 根据用户给出的空间位置坐标或目标星源名称, 检索并访问全球众多天文项目的天文数据; ②Iris (SED 分析工具): 采用该桌面应用软件来检索、绘制光谱能量分布图 (spectral energy distribution, 简称 SED); ③时序检索工具: 从三大主要数据归档系统中检索时序数据, 并采用 NASA 系外行星归档周期图分析时序数据; ④交叉认证工具: 在用户输入的星表与通用天文星表之间执行快速高效的交叉认证, 输入星表中的星源条目最高可达一百万; ⑤JAVOT: Java 语言开发的 VOTable 解析器; ⑥VOClient: 访问 VO 环境中各天文数据集的桌面应用软件; ⑦STSCI 注册系统等。此外, VAO 对 VO 概念的宣传推广做出了大量工作, 并承担了 IVOA 联盟内部许多协调管理工作。

(2) 英国虚拟天文台

英国天文网格 (AstroGrid) 项目由英国粒子物理与天文研究委员会 (PPARC) 资助, 于 2001 年 9 月启动, 耗资共一千四百万英镑, 是英国 e-Science 计划网格项目之一。其技术目标是: ①与 IVOA 成员一同开发元数据及数据交换标准; ②建立可供 AstroGrid 与各大数据中心共享的物理网格资源, 提供软件开发基础设施; ③基于天文学家的应用需求搭建虚拟天文台系统, 整合英国各大主流天文数据库, 建立和维护资源注册、公共存储、数据访问、处理分析等服务; ④通过自主研发或者技术引入等方式为用户提供一套 VO 环境下可交互访问各类数据资源的科学应用工具。

AstroGrid 项目通过多年的推进与执行, 取得了巨大的科研成果。在应用软件方面, 已研发出如下产品: ①VODesktop 桌面应用软件, 用户首先选择已成功注册的数据资源, 在该数据资源中检索、访问所需数据, 下载、保存数据于本地或公共存储空间中, 也可将数据导入到其他应用软件中, 如 Aladin 可视交互软件、VOSpec 光谱分析工具等, 该软件界面如图 1.5 所示; ②Topcat 天文可视化

分析工具,可浏览、分析 FITS 和 VOTable 格式的天文数据,并提供画图、统计、计算、交叉认证等功能;③Astro Runtime 中间件,为用户提供可调用 VO 服务的 API,辅助构建工作流、VO 交互工具等。

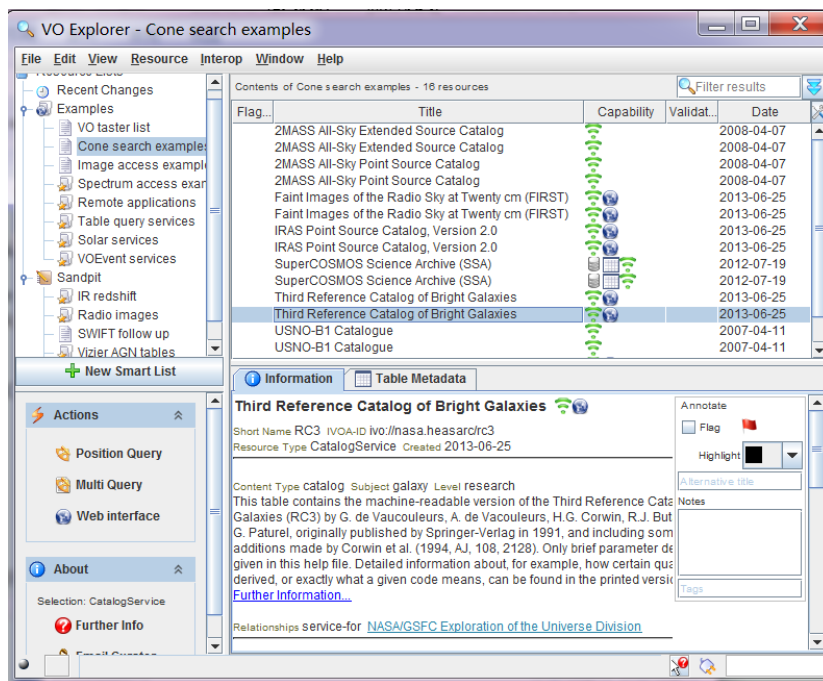


图 1.5 VODesktop 应用软件

(3) 中国虚拟天文台

自 2002 年起,中国科学院国家天文台正式开始构建中国虚拟天文台(China-VO),并加为 IVOA 成员。China-VO 的重点研究领域包括如下几个方面:支持 VO 的天文项目与观测设施、国内外天文研究资源的统一访问、中国虚拟天文台系统平台的开发、基于 VO 的天文研究示范、基于 VO 的天文科普教育^[16]。

在软件开发方面,China-VO 从起初研制小型工具软件和中型应用程序,逐渐向开发大型天文应用平台过渡。已成功研制的软件工具及服务包括:①VOFilter, VOTable 转换插件;②VOImpat, 可进行星表联合查询的可视分析工具;③FitHAS, 可批量读取 FITS 头信息并导入关系型数据库的软件工具;④VO-DAS(Virtual Observatory Data Access Service)VO 数据访问服务;⑤SkyMouse 天文数据检索门户,如图 1.6 所示。



Object Name	
	Examples: vega, M 31, 12 30 45 +10 20, 1999ApJS...125..409C
Region size(degrees)	0.25
UserType	select
Services List	
Information	☒ Simbad
Catalog	☒ VizieR Catalogues
Sky Atlas	☒ DSS Service
Bibliography	☒ CDS Bibliographic
	☒ Anxiv Service
	☒ ADS Service

Other Service you can select below:

图 1.6 SkyMouse 检索界面

1.3 研究内容

本文在广泛调研国际上天文数据检索发布系统发展现状的基础上, 基于虚拟天文台提供的协议标准与技术框架, 构建了符合 VO 标准规范的 HXMT 卫星数据检索发布系统, 并基于 J2EE 技术中的 Spring 框架、采用 Web Service 开发技术进行系统研发与实现。具体内容如下。

1、虚拟天文台基础理论与技术框架的研究

对虚拟天文台的产生背景、概念定义、制定的标准协议、以及国内外现阶段的发展情况展开深入研究, 为将 VO 技术框架引入 HXMT 项目奠定理论基础。首先, 从国际上天文数据检索发布系统的发展趋势着手, 分析引入 VO 技术的必要性, 并对各国虚拟天文台的发展情况及应用形式进行充分调研; 然后, 归纳总结常用的 VO 标准协议, 重点涵盖体系结构、数据格式和数据访问协议三个方面。最后进一步明确本系统所选用的 VO 标准协议。

2、HXMT 卫星数据组织管理

通过进行数据产品分级, 设计归档目录结构, 确定数据存储格式以及数据产品种类等, 对 HXMT 卫星数据进行合理高效的归档管理。

3、HXMT 卫星数据检索发布系统设计

基于对虚拟天文台标准协议及应用服务的研究分析, 进行 HXMT 卫星数据检索发布系统的原型设计。首先密切结合系统需求, 明确功能模块; 然后参考

IVOA 体系结构，构建一套符合 VO 标准规范的体系架构；最后，在详细研究分析 HXMT 卫星数据的基础上，设计数据模型来管理和使用数据。通过对功能模块、体系架构和数据模型进行系统性分析与设计，为后续的系统研制与实现做好充分准备。

4、HXMT 卫星数据检索发布系统实现

研制并实现基于 VO 框架的 HXMT 卫星数据检索发布原型系统。首先，确定系统开发环境，包括开发语言、开发运行环境、Web 服务容器、数据库等，并针对开发过程中需采用的应用技术进行简要选取和分析。然后，分模块编写代码实现系统功能，并选取合适的索引算法优化数据库结构以提升检索性能。最后，展示原型系统各功能模块的运行结果。

1.4 本文组织结构

论文共分五章，各章的主要内容描述如下：

第一章简要介绍了本课题的研究背景及意义，在充分调研天文数据检索发布系统发展历程的基础之上，逐步引出虚拟天文台的概念，并分析国内外 VO 的发展现状，最后归纳总结了本课题的研究内容。

第二章先具体阐述了虚拟天文台的标准协议与规范，包括 VO 体系结构、VOTable 数据格式、VO 数据访问协议。然后对系统研制过程中采用的计算机技术及应用模式进行简要介绍，包括 SOA 架构、MVC 模式和 Spring 框架。

第三章在分析系统需求及设计目标的基础之上，确定了 HXMT 卫星数据检索发布系统的主要功能模块，并基于 VO 技术框架，建立了一套符合 VO 标准的体系架构。同时，针对 HXMT 数据产品特点，进行数据模型设计、数据库设计、数据库索引设计，为后续的系统研制与实现做好充分准备。

第四章首先基于 SOA 架构的思想，采用 Spring 框架和 Web Service 技术进行系统研制与 Web 服务开发；随后详细阐明了检索浏览、数据下载、可视交互三大功能模块的实现方法，以及建立 HTM 数据库索引的具体步骤，并且成功验证了锥形检索服务的 VO 有效性；最后依次展示了各功能模块的运行效果。

第五章对全文内容进行总结，并对进一步的工作提出了展望。

第二章 虚拟天文台基础理论与应用技术研究

2.1 虚拟天文台标准与协议

2.1.1 体系结构

1、IVOA 概念体系结构

IVOA 定义的 VO 概念体系结构采用了分级的描述方法, 共分为 0 级、1 级和 2 级三级。下面从粗粒度到细粒度渐进地展开讨论。0 级是在抽象层面上提供简单、通俗的架构描述, 仅限于划分层次和定义每一层次的基本功能, 如图 2.1 所示。1 级在 0 级描述的基础上提供了稍多的组成元素和功能细节, 指出每一层应用到的服务或数据模型, 并分析各层之间的通信交互方式, 但尚未引入具体技术细节, 如图 2.2 所示。2 级将各项 VO 标准协议融入到 VO 概念体系结构中, 明确每一项服务、每个功能点所需用到的标准规范与协议, 如图 2.3 所示。

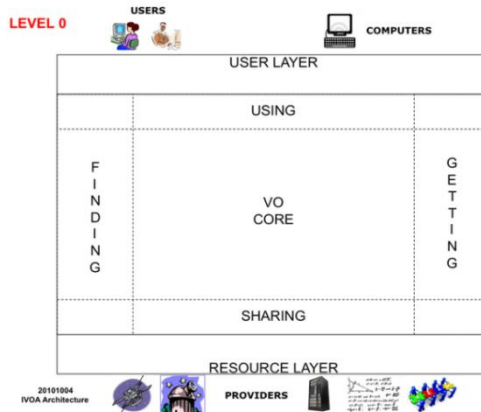


图 2.1 0 级 VO 概念体系结构

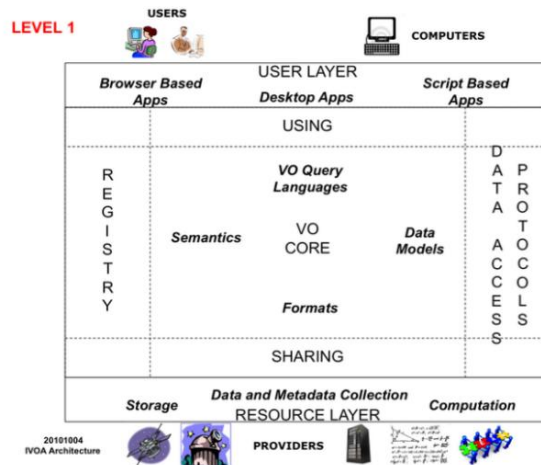


图 2.2 1 级 VO 概念体系结构

图 2.3 2 级 VO 概念体系结构

资源层是天文数据的提供者，可为用户提供数据集（图像、光谱、星表、时序数据、理论模型、模拟计算结果等）及其相关的描述元数据，存储设施和计算设施等^[17]。

用户层是天文数据的消费者，科研人员和天文爱好者可直接通过用户层提供的单机版软件、Web 网页或脚本程序使用服务层的各类服务，进而访问获取底层数据资源。

英国 AstroGrid 项目构建了一个较为成功的 VO 体系结构,并完成了整个 VO 体系结构各层次的开发。该体系结构的组件图如图 2.4 所示。其中, Portal 属于用户层, 图像、星表和程序数据属于资源层, 其余组件属于服务层。

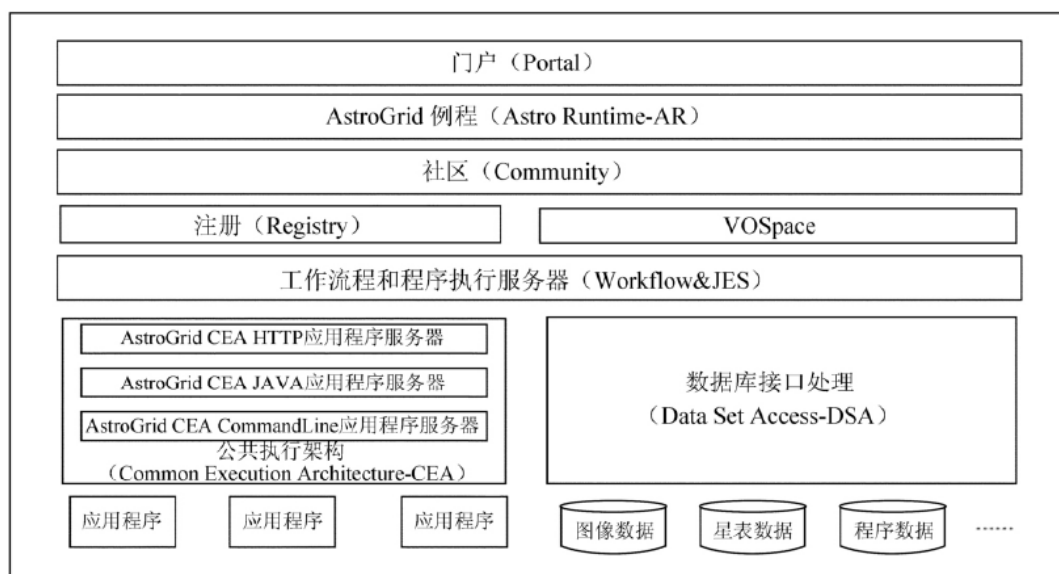


图 2.4 AstroGrid 组件

Portal 为与用户直接交互的 Web 界面，封装了 AstroGrid 服务层提供的各类服务。

Community 是用于创建并管理单点登录认证的 Web 应用，提供用户授权、权限管理、用户信息管理等功能。

Regsitry 提供符合 IVOA 标准的注册服务，用于收割数据和发布数据。

VOSpace 是用于创建虚拟文件存储系统的可部署服务，为用户提供类似网盘的在线数据存储区，并定义了远程存储标准。该标准已成为 IVOA 标准协议之一。

Workflow 通过 Python 脚本提供自动化任务处理流程。而 JES 作为工作流引擎，可处理复杂的工作流，用户可通过 JES 接口安排、提交、监测任务，如搜索注册表、异步查询星表等^[18]。

DSA 组件提供既符合 SCS、TAP 等 VO 标准协议、又可访问各类关系型数据库的统一访问接口，属于 VO 标准协议与异构数据库中各种数据格式之间的转换插件。

CEA 作为中间件，提供标准的参数传递机制和 Web 接口。CEA 向下封装各类已发布的应用程序，包括命令行应用程序、基于 HTTP 协议的 Web 应用程序、Java 编写的应用程序等；向上屏蔽已注册服务的底层实现细节，从而使得 JES 工作流系统不需要针对不同的服务应用编写各类代码，即可驱动相应的数据检索访问、数据格式转换等服务^[18]。

3、中国 China-VO 体系结构

中国 China-VO 体系结构的层次模型如图 2.5 所示，共分为构造层、资源层、汇集层、用户层四层。

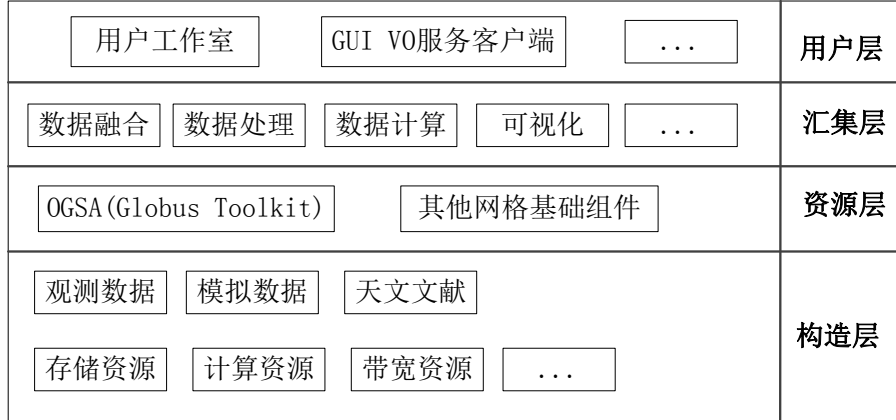


图 2.5 China-VO 体系结构

构造层为 China-VO 系统提供了丰富的数据资源以及完备的软硬件设施，包括各种天文数据资源（星表、光谱、图片等）、计算资源（计算机、服务器等）、网络资源（路由、带宽等）、存储资源（硬盘、磁带等）等^[19]。

资源层采用符合开放网格服务架构（Open Grid Services Architecture，OGSA）的网格系统，提供统一的数据访问、计算访问、网格系统管理等功能。

汇集层是整个 VO 的核心，集成了丰富的 VO 组件以解决数据访问、注册、获取等问题，提供了数据融合、统计分析、可视化、数值计算等服务。

用户层直接与用户交互，用于接收用户请求并返回处理结果。

综上分析可知，英国 AstroGrid 重点对 IVOA 概念体系架构进行裁剪，实现了相对较为完整的、符合 VO 规范的体系架构。中国 China-VO 体系架构则明确采用网格技术作为底层屏蔽物理差异的基础设施^[20]，并封装各类服务提供给用户。通过充分调研 IVOA 概念体系架构，并对比分析中英两国在各自项目中设计 VO 体系架构的具体实例，最终决定 HXMT 卫星数据检索发布系统在遵循 IVOA 提出的 VO 概念体系结构的前提下，根据自身需求和数据特性，选择性支持 VOTable 数据格式、UCD 语义标准和 SCS 数据访问协议，构建一套由资源层、服务层、用户层组成的 VO 体系结构，以便实现与其他天文项目数据集之间的互访问、互操作。

2.1.2 VOTable 数据格式

众所周知，普适图像传输系统(Flexible Image Transport System, 简称 FITS)是天文学领域通用的数据存储格式标准^[21]。绝大多数天文数据均以 FITS 文件的形式存储于计算机文件系统中。但 FITS 也具有一些缺点，例如 FITS 头的元数据难以提取并应用，FITS 无法流式化来实时响应查询，FITS 元数据和数据本身无法分离，不支持 Unicode 编码，不支持 UCD 描述表列和参数的语义内容等。因此，为了实现分布式异构数据集之间的互访互操作并满足高效网络传输的需求，IVOA 团队制定了 VO 标准数据格式——VOTable。

VOTable 是一种 XML 数据格式，用于封装数据以便于异构系统间数据的互访与传输^[22]。VOTable 数据模型如图 2.6 所示，包含头部描述表列属性的元数据和真实表列数据两大部分。

```

VOTable = hierarchy of Metadata + associated TableData, arranged as a set of Tables
Metadata = Parameters + Infos + Descriptions + Links + Fields + Groups
Table = list of Fields + TableData
TableData = stream of Rows
Row = list of Cells
           Primitive
Cell = or variable-length list of Primitives
           or multidimensional array of Primitives
Primitive = integer, character, float, floatComplex, etc

```

图 2.6 VOTable 数据模型

VOTable 能灵活且高效地传输数据，支持传输 TABLEDATA、FITS、BINARY 三种格式的数据。最为典型的数据承载方式为 TABLEDATA，主要用于系统间星表数据的互访互操作，如图 2.7 所示。

```

<?xml version="1.0"?>
<VOTABLE version="1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >
  <RESOURCE name="myFavouriteGalaxies">
    <TABLE name="results">
      <DESCRIPTION>Velocities and Distance estimations</DESCRIPTION>
      <GROUP ID="J2000" utype="stc:AstroCoords">
        <PARAM datatype="char" arraysize="*" ucd="pos.frame" name="cooframe"
          utype="stc:AstroCoords.coord_system_id" value="UTC-ICRS-TOP0" />
        <FIELDref ref="col1"/>
        <FIELDref ref="col2"/>
      </GROUP>
      <PARAM name="Telescope" datatype="float" ucd="phys.size;instr.tel"
        unit="m" value="3.6"/>
      <FIELD name="RA" ID="col1" ucd="pos.eq.ra;meta.main" ref="J2000"
        utype="stc:AstroCoords.Position2D.Value2.C1"
        datatype="float" width="6" precision="2" unit="deg"/>
      <FIELD name="Dec" ID="col2" ucd="pos.eq.dec;meta.main" ref="J2000"
        utype="stc:AstroCoords.Position2D.Value2.C2"
        datatype="float" width="6" precision="2" unit="deg"/>
      <FIELD name="Name" ID="col3" ucd="meta.id;meta.main"
        datatype="char" arraysize="8"/>
      <FIELD name="RVel" ID="col4" ucd="spect.dopplerVeloc" datatype="int"
        width="5" unit="km/s"/>
      <FIELD name="e_RVel" ID="col5" ucd="stat.error;spect.dopplerVeloc"
        datatype="int" width="3" unit="km/s"/>
      <FIELD name="R" ID="col6" ucd="pos.distance;pos.heliocentric"
        datatype="float" width="4" precision="1" unit="Mpc"/>
      <DESCRIPTION>Distance of Galaxy, assuming H=75km/s/Mpc</DESCRIPTION>
    </FIELD>
    <DATA>
      <TABLEDATA>
        <TR>
          <TD>010.68</TD><TD>+41.27</TD><TD>N 224</TD><TD>-297</TD><TD>5</TD><TD>0.7</TD>
        </TR>
        <TR>
          <TD>287.43</TD><TD>-63.85</TD><TD>N 6744</TD><TD>839</TD><TD>6</TD><TD>10.4</TD>
        </TR>
        <TR>
          <TD>023.48</TD><TD>+30.66</TD><TD>N 598</TD><TD>-182</TD><TD>3</TD><TD>0.7</TD>
        </TR>
      </TABLEDATA>
    </DATA>
  </TABLE>
</RESOURCE>
</VOTABLE>

```

图 2.7 VOTable 实例

其中，需重点介绍的内容为统一内容描述符（Unified Content Descriptors，简称 UCD）。UCD 为 VOTable 表中的每列提供唯一的标识符，例如“赤经”这一属性在不同系统中可描述为“RA”或“alpha”等自定义名称，当系统间采用 VOTable 进行数据互访时约定“赤经”对应的 UCD 必须标识为“pos.eq.ra”，如此便屏蔽了表列命名及访问位置的差异性。因此，当应用程序查询不同的天文项目中的数据资源，并处理返回的多个 VOTable 结果数据时，UCD 使得对检索结果进行逻辑合并成为可能，从而方便应用程序定位查找所需数据信息。

目前，国际上已成功开发了 Java、C++、Perl 语言的 VOTable 解析包，用于读、写、修改 VOTable 数据文件，方便程序调用。如图 2.8 所示。

	名称	描述	链接
Java	JAVOT	读	http://www.us-vo.org/VOTable/JAVOT/
	SAVOT	读/写/修改	http://cdsweb.u-strasbg.fr/devcorner.gml
	STIL	读/写/修改	http://www.starlink.ac.uk/stil/
	VOTable Java Parser based on XML Schema	读/写	http://spider.ipac.caltech.edu/staff/jchavez/public/votable_parser.html
	VOTable Java Streaming Writer	写	http://vo.iucaa.ernet.in/~voi/votableStreamWriter.htm
C++	C++ Parser	读	http://vo.iucaa.ernet.in/~voi/cplusplusparser.htm
Perl	VOTable Perl Modules	读/写	http://heasarc.gsfc.nasa.gov/classx/votable/
	VOTable::DOM	格式/打印	http://trac.us-vo.org/project/nvo/wiki/VOTable-DOM

图 2.8 VOTable 解析包

VOTable 采用 XML 能够充分利用行业内标准的工具和软件，具有良好的可

移植性^[2]；VOTable 可进行数据流整合、过滤等处理，方便远程调用和网络传输；VOTable 提供了丰富的元数据描述机制，实现了元数据与数据分离存储，这对数据检索的高效实现具有重要作用；同时，VOTable 为异构系统间数据的互访问互操作提供了标准的数据格式，有利于实现跨平台跨区域系统间的数据共享。

2.1.3 数据访问协议

1、简单锥形检索（Simple Cone Search，简称 SCS）协议

应用锥形检索（SCS）协议可获取某一给定天区位置周围的数据信息。锥形检索是指给定天球上一位置坐标 (α, δ) 和检索半径 (r) ，找出与 (α, δ) 的角距离小于 r 的所有表格化的天体目标^[23]。相当于以赤道坐标系的原点——太阳作为锥体顶点，以太阳到 (α, δ) 的连线为中心线，以 r 为锥顶角，形成一个无底的圆锥体。通过锥形检索即可以查找到无底圆锥体内包含的所有天体。如图 2.9 所示。

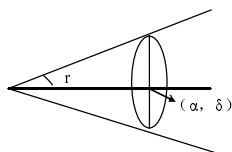


图 2.9 锥形检索概念图

SCS 协议在技术实现上多以 Web 服务方式呈现，相应的 SCS 服务需遵循如下接口约束。

(1) 必须以标准 URL 方式发送 HTTP GET 请求，标准 URL 方式为 `http://<server-address>/<path>?[<extra-GET-arg>&[...]]`，其中，server-address 代表服务部署的域名；path 代表本地路径；extra-GET-arg 为携带的参数键值，如 `<name>=<value>`，其中 name 有三个参数，为 RA 赤经，DEC 赤纬，SR 角距离。URL 举例如下：

`http://mycone.org/cgi-bin/search?RA=180.567&DEC=-30.45&SR=0.0125`

(2) 必须以 VOTable 格式的 XML 文件返回检索结果，HTTP 响应的媒体类型为 text/xml。

(3) 若检索服务出现错误，则必须返回携带“Error”错误反馈信息的 VOTable 文件。

2、简单图像访问（Simple Image Access，简称 SIA）协议

简单图像访问（SIA）协议可用来获取指定天空区域内的图像数据。SIA 服务约束条件如下。

（1）必须以标准 URL 方式发送 HTTP GET 请求，标准 URL 方式为 `http://<server-address>/<path>?[<extra GET arg>&[...]]`。较之于 SCS 协议，SIA 的 URL 地址中可设置更多参数，如图像大小、图片格式、坐标系、旋转角度等。

（2）必须以 VOTable 格式的 XML 文件返回检索结果，比 SCS 约束条件更多。例如，Image Metadata 必须包含的域有：

```
ucd="VOX:Image_Title", with datatype="char", and arraysize="*";
ucd="POS_EQ_RA_MAIN" , with datatype="double" ;
ucd="POS_EQ_DEC_MAIN", with datatype="double;
ucd="VOX:Image_Naxes", with datatype="int"; ucd="VOX:Image_Naxis",
with datatype="int", and arraysize="*";
ucd="VOX:Image_Scale", with datatype="double", and arraysize="*";
ucd="VOX:Image_Format", with datatype="char", and arraysize="*";
```

Access Metadata 必包含域 `ucd="VOX:Image_AccessReference"`，with `datatype="char"` and `arraysize="*"`，其中通过图片链接 URL 可以直接获取到真实图片。

3、表访问协议（Table Access Protocol，简称 TAP）

表访问协议（TAP）定义了用于访问表数据的服务协议。表包括天文星表、数据库表等。可访问数据包括数据库表数据、表列元数据和真实表数据。该协议支持同步和异步查询，并提供简单的交叉认证功能^[2]。

4、天文数据查询语言（Astronomical Data Query Language，简称 ADQL）

ADQL 是针对天文数据的一种查询语言。可通过 Open SkyQuery Portal 门户网站输入 ADQL 语句进行天文数据库查询，检索天文星表或其它形式存在的数据集，完成锥形检索和交叉认证^[2]。该标准语言是两种重要查询服务的核心：SkyNode 和 Open SkyQuery Portal。SkyNode 是单站点单一星表查询的标准服务接口，Open SkyQuery Portal 是利用 ADQL 在所有已知 SkyNode 服务间实现跨越

式查询的服务^[37]。ADQL 的一大优势是可以方便实现交叉证认,例如, XMATCH (s, t, !o) <2.5, 即可检索呈现出现在前两表而未出现在第三个表, 且在前两表中是同一对象的卡方概率小于 2.5 的所有星源。

2.2 应用技术

2.2.1 SOA 架构

面向服务的体系结构(Service-Oriented Architecture, SOA)是一个组件模型, 将应用程序的不同功能单元(称为服务)通过定义良好的接口和契约联系起来^[24]。接口完全独立于实现服务的任何操作系统、编程语言或硬件平台^[25], 从而使得不同系统中的各类服务可以以一种统一和通用的方式进行交互^[26]。

SOA 架构中应用到的 Web 服务协议如图 2.10 所示。在 Web Service 协议栈中, 从底层向上依次是网络传输层、消息传递层、服务描述层、服务发布层、服务发现层。其中, 网络传输层作为 Web Service 协议栈的基础, 包含 HTTP、FTP 以及 SMTP 等多种网络协议。消息传递层使用 SOAP(简单对象访问协议)作为消息传递协议, 以实现服务提供者、服务注册中心和服务请求者之间进行信息交换。服务描述层使用 WSDL(Web 服务描述语言)作为消息协议, WSDL 使用 XML 语言来描述网络服务, 能提供 Web 服务的一些特定信息, 包括功能、接口、结构等定义与描述。服务发布层和服务发现层均使用 UDDI(统一描述、发现和集成技术协议)作为服务的注册与发现协议。UDDI 提供了 Web 服务的注册库, 用于存储 Web 服务的描述信息, 同时, 为服务请求者提供查询、绑定的功能^[27]。

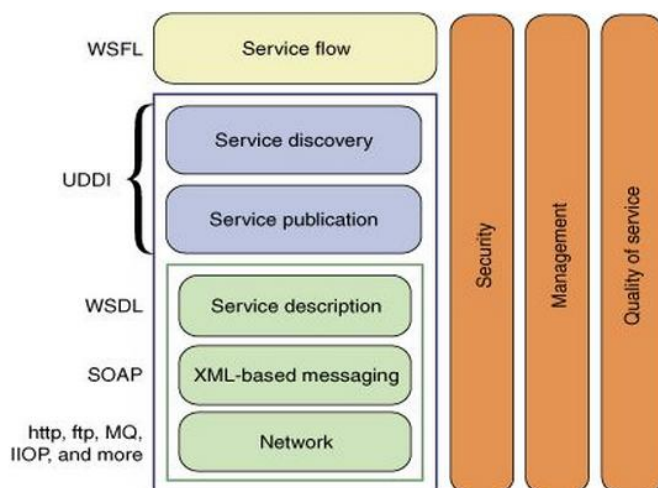


图 2.10 Web 服务协议栈

Web 服务调用流程如图 2.11 所示。Web 服务提供者首先使用 WSDL 描述自己的服务接口，通过使用 UDDI 在服务发布层向服务注册中心提交 Web 服务描述、发布 Web 服务接口。当 Web 服务请求者对服务注册中心查找 Web 服务时，服务发现层返回相应的 WSDL 描述。通过 WSDL 文档的描述即可绑定相应的服务接口，从而成功调用请求的 Web 服务，实现 Web 服务请求者和 Web 服务提供者之间的交互。

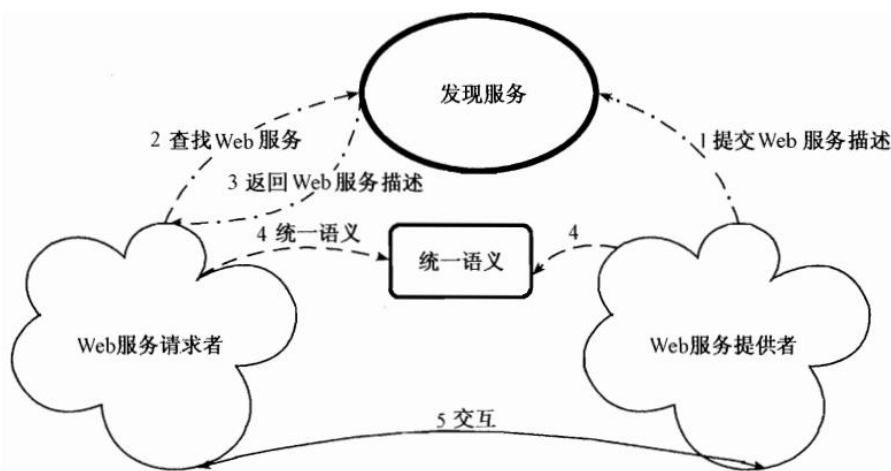


图 2.11 Web 服务调用流程

SOA 架构作为发布 Web 服务的基础架构，应用广泛，分层清晰，每层均定义了规范化的 Web 服务标准，而且具有互操作性、松耦合、可复用等一系列优点。因此，在研制 HXMT 卫星数据检索发布系统的过程中，基于 SOA 架构开发 Web 服务。例如，按照 SCS 协议将锥形检索功能封装为锥形检索服务，以 Web 服务的方式暴露出来。该 Web 服务不仅可为本系统的上层应用提供锥形检索功

能,也可被其他采用 SOA 架构的异构系统所调用,从而使得系统构建更为灵活、轻便,也有利于屏蔽异构系统之间的差异性,实现跨平台的互访问、互操作。

2.2.2 MVC 模式

MVC (Model-View-Controller, 模型-视图-控制器)是在 20 世纪 80 年代为编程语言 Smalltalk-80 提出的一种软件设计模式,当时主要被用来构建 GUI 界面^[28],目前已被广泛应用于 Web 系统的设计与研制过程中。MVC 强制性地将应用程序分成模型、视图、控制器三部分,各部分的功能如下:

1、模型负责处理整个应用程序中的业务逻辑,包括数据库交互、执行具体算法等。模型的作用如下:(1)抽象出问题领域中的对象,封装对象的属性及各类操作方法;(2)向控制器提供对业务逻辑的访问;(3)接受视图的数据查询和增删改请求,负责与底层数据库的访问与交互;(4)当数据发生任何变化时,主动通知相关视图进行更新^[29]。

2、视图是直接与客户交互的界面,用于接收用户输入的数据、更新用户请求,并将接收数据传输给控制器。也可用来显示模型处理后的数据结果。

3、控制器一方面负责分析用户请求,调用合适的模型来处理响应请求,另一方面又负责选择合适的视图来展示业务处理结果,在模式和视图之间扮演着组织者的角色。

应用 MVC 模式进行系统研制主要拥有如下优点:

1、增强代码复用性

多个视图共享一个模型,提升了模型组件的复用率。前端视图技术种类繁多,包括 HTML 网页、JSP 页面、单机版软件 GUI 界面等。由于模型返回的数据并未进行特定的格式化,因此不同技术开发的多个前端视图,均能显示同一个模型返回的数据,极大地增强了模型组件的复用性^[30]。同时,模型变动时不会影响视图显示,只需修改模型即可,有利于统一管理模型。

2、实现松耦合

由于开发系统的模型、视图、控制器之间相对独立,所以在系统研制或者软件维护过程中,若遇到需求变更、系统升级等情况,可以只将改动一小部分,完全不需要牵一发而动全身。例如,模型中的处理算法改动不会影响到模型与控制

器以及视图之间的接口，进而也不会影响视图显示。由此可见，MVC 模式充分实现了松耦合效果，使软件更易维护管理。

2.2.3 Spring 框架

MVC 模式的广泛应用，催生了 Spring 框架。本文选用轻量级 Spring 框架来实现 MVC 模式。Spring 框架是于 2003 年兴起的一个轻量级、开源 Java 开发框架，其核心思想来源于 Rod Johnson 编写的著作 Expert One-On-One J2EE Development and Design^[31]。Spring 框架采用基本的 JavaBean 替代重量级的 EJB 技术，简化 JavaEE 的企业级应用开发，以轻量级、低侵入性的框架来满足大多数项目的要求，使得开发人员可以集中精力于业务逻辑层，无需关心特定的部署环境。Spring 的特色包括依赖注入、面向切面编程、MVC 实现、RESTful 和 SOAP 样式的 Web 服务开发、支持 JDBC 等。

Spring 框架是由 7 个定义良好的模块组成的分层架构^[32]，如图 2.12 所示。基于 Spring 框架开发软件系统过程中，可根据项目需求独立实现整个 Spring 框架中的某一模块，也可将多个模块联合实现。

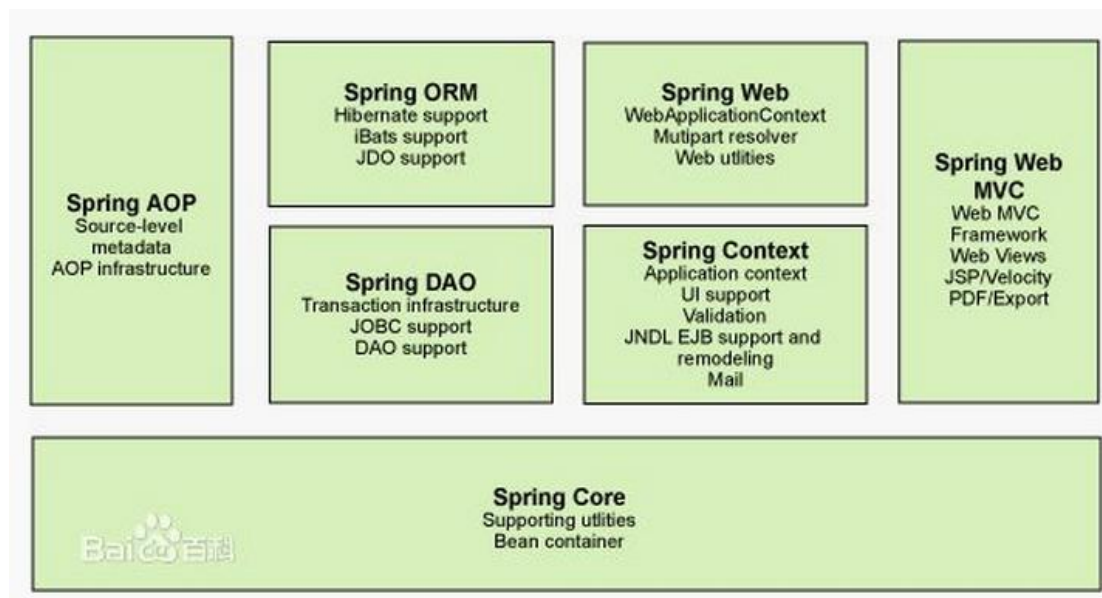


图 2.12 Spring 框架

每个模块的功能如下：

1、核心容器：提供 Spring 框架的基本功能，其中，BeanFactory 作为主要组件使用控制反转（Inversion of Control, IOC）原则将配置与实际的应用程序代

码分开，实现了工厂模式。

2、Spring 上下文：通过配置文件，提供应用上下文信息、UI 支持、JNDI EJB 支持、邮件服务、校验和调度功能。

3、Spring AOP：通过配置管理实现了面向切面的编程功能，不依赖于 EJB 组件，即可在应用程序中集成声明性事务管理^[33]。

4、Spring DAO：JDBC DAO 抽象层提供了清晰明确的异常等级规范，从而简化了错误处理过程，减少了异常处理的编码量。

5、Spring ORM：Spring ORM 模块遵循 Spring 事务处理和 DAO 异常等级规范，以事务策略、资源容器、DAO 实现等形式集成到 Hibernate、JDO、iBATIS 框架中^[34]。

6、Spring Web：为 Web 应用程序提供了上下文信息，且支持与 Struts 的集成^[35]。

7、Spring Web MVC：提供基于核心容器、灵活、松耦合且扩展性极强的 MVC Web 应用程序框架。该框架通过策略接口变得高度可控，且容纳多种视图技术^[36]，例如 JSP、Velocity 等。Spring 中间层可与许多 MVC 模式的表示层框架完成无缝集成，如 Struts、WebWork 等。

2.3 本章小结

本章着重介绍了虚拟天文台的标准与协议，包括 VO 概念体系结构、数据互操作格式 VOTable、天文数据查询语言 ADQL、简单锥形检索协议 SCS、简单图像访问协议 SIA 等。同时，针对系统开发过程中应用到的计算机技术与应用框架进行了概括描述，包括 SOA 架构、MVC 模式、Spring 框架。在系统研制过程中会参考 IVOA 概念体系结构设计自身的体系架构，采用 VOTable 数据格式与 Aladin 等 VO 兼容软件进行可视交互，并遵循 SCS 数据访问协议开发 Web 服务以提供锥形检索功能。在技术实现方面，会采用 SOA 架构来开发 Web 服务，采用 Spring 框架来实现 MVC 模式，以达到松耦合效果。本章为后续系统的设计与实现奠定了理论基础。

VO 标准协议为不同天文项目之间实现数据的互访互操作提供了可能。理论

上,只要各国的天文项目遵循 VO 标准协议与规范进行数据检索发布系统的研制与开发,经注册之后不同项目的数据资源就可实现无缝链接。几乎所有的虚拟天文台项目都希望参与讨论和制订 VO 互操作标准,同时,遵循 VO 标准协议来开发应用系统,以融入国际虚拟天文台的大家庭。HXMT 卫星数据检索发布系统也不例外,后面会详细阐述如何在系统研制过程中引入 VO 标准协议与技术框架。

第三章 系统设计

3.1 系统设计目标

HXMT 卫星数据检索发布系统的设计目标包括：

- 1、对 HXMT 卫星数据进行分级定义，以合理的目录组织结构归档存储数据，并设计合适的数据模型来管理和使用数据。
- 2、设计用户友好型的数据检索发布系统，将物理特性复杂、数据种类繁多的 HXMT 卫星数据发布出去，支持用户对 HXMT 各级数据产品的检索访问与浏览下载，为我国空间天文卫星数据检索发布系统的研制奠定基础。
- 3、遵循 VO 标准协议进行系统研制，将 HXMT 数据融入 VO 环境中，从而实现与其他天文项目数据资源之间的互访问、互操作。

3.2 系统功能设计

HXMT 卫星数据检索发布系统的主要功能包括 HXMT 卫星各级数据产品的汇交、推送、发布、检索浏览、数据下载、可视交互、用户管理、状态监控、授权管理等。下面重点对检索浏览、数据下载、可视交互三大核心功能模块进行介绍。

1、检索浏览

检索浏览模块主要提供对 HXMT 卫星各级数据产品的检索访问和数据显示功能。按检索条件可分为锥形检索、组合检索两种检索访问方式。其中，锥形检索是指基于赤经、赤纬和检索半径共三项限定条件进行的检索，用户提交包含目标天体位置坐标和检索半径信息的请求后，即可检索出与目标位置的角距离小于检索半径的所有表格化的天体目标^[23]，并返回 VOTable 数据格式的检索结果；组合检索是指多重条件组合检索，针对 HXMT 卫星数据的特性，设置更为详尽的检索条件，如观测号、提案号、载荷名称、观测模式、观测起始时间、有效曝光时间、数据发布状态、数据发布时间。数据显示功能负责在检索操作后，根据用户权限显示符合检索条件的数据列表，并可实现升/降序排序。

2、数据下载

数据下载模块主要提供数据导出和打包下载功能。其中，数据导出是指用户可将检索结果数据导出为 VOTable、CSV、HTML、XSL、TXT 数据格式文件；打包下载是指用户在检索结果显示界面勾选数据确定待下载数据条目后，即可进行单条 FITS 格式数据或批量数据的打包下载。

3、可视交互

可视交互模块集成了 Aladin 开源软件，提供关联检索、点源标注、可视浏览三个功能。（1）关联检索：Aladin 接收从检索结果显示界面提取的目标名称或赤经赤纬位置信息，自动检索 Simbad 数据库、SDSS 巡天等天文数据项目对应位置的数据，并将数据以点源显示、图片加载、星表条目自动添加的方式呈现出来^[37]。（2）点源标注：将检索结果显示界面检索结果数据以 VOTable 格式导入到 Aladin 中，Aladin 会自动将 VOTable 中的赤经、赤纬信息以点源方式标注在 Aladin 界面上，以方便用户直观地查看检索结果显示界面数据在天球中的位置。（3）可视浏览：将检索结果显示界面中的 FITS 格式图片文件导入到 Aladin 中进行展示，方便用户浏览具体的 FITS 格式图片内容。

3.3 体系架构设计

参照 IVOA 定义的 VO 概念体系结构^[38]，并结合实际应用需求与项目自身特性，对 HXMT 卫星数据检索发布系统进行体系架构的设计。本系统的体系架构分为资源层、服务层、用户层三层，如图 3.1 所示。

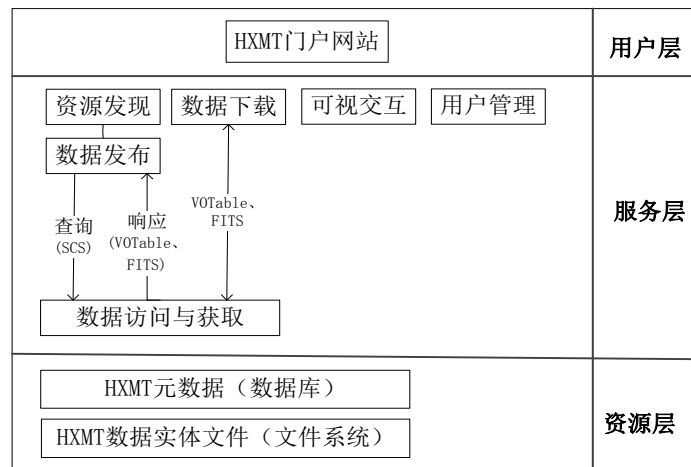


图 3.1 HXMT 卫星数据检索发布系统的体系架构

(1) 资源层负责存储 HXMT 卫星科学数据和辅助数据, 为上层提供数据资源。主要包括数据实体文件和元数据两大部分。数据实体文件多存储于文件系统中, 包括可面向用户发布的 HXMT 卫星 1 级科学数据产品和 2 级科学数据产品。元数据是对数据实体的特征属性提供结构化描述的数据, 包括从物理数据实体文件中提取的观测号、赤经、赤纬、提案号、载荷名称、观测模式、观测起始时间、有效曝光时间、数据发布状态、数据发布时间等数据信息以及文件大小、文件路径等文件信息。元数据信息主要存储于关系型数据库中, 以便于上层检索。

(2) 服务层作为整个系统的核心, 集成了丰富的 VO 服务组件, 主要提供以下 3 类服务。

面向数据资源, 提供数据访问与获取服务。数据访问与获取服务直接与资源层进行交互, 对资源层中的数据资源进行快速访问与获取, 同时, 为上层屏蔽底层实现细节, 提高系统的透明性。

面向应用需求, 提供数据发布、资源发现、数据下载、可视交互服务。数据发布是指通过调用数据访问与获取服务, 开发符合简单锥形检索 (SCS) 协议的 Web 服务接口, 响应上层查询请求并将检索结果封装为标准的 VOTable 格式数据, 从而将 HXMT 数据资源暴露给用户的过程。资源发现服务用于接收并处理用户层的检索请求, 返回符合检索条件的数据以供用户浏览。数据下载服务既可根据用户层的下载请求, 下载 FITS 格式保存的 HXMT 卫星数据实体文件, 也可将检索结果导出为 VOTable 格式文件。可视交互服务负责将检索结果数据导入到 Aladin 开源软件中, 进行关联检索、点源标注和 FITS 图片显示, 以便用户分析研究。

面向用户, 提供用户管理服务。根据项目需求, 用户主要分为数据管理员、科学用户、公众用户三类角色, 不同类别的用户拥有不同层次的权限。依据用户角色的不同, 用户管理服务为用户设置不同的可访问数据范围及访问深度, 并实现用户身份认证、用户访问权限授予、服务使用权限限制、注册用户信息管理等功能。

(3) 用户层主要负责直接与用户进行交互, 提供检索浏览 HXMT 卫星数据的 Web 门户网站。用户层通过调用服务层集成的服务, 实现检索浏览、数据下

载、可视交互等应用功能。

3.4 数据模型设计

3.4.1 数据产品分析

1、HXMT 科学数据产品

(1) 观测模式

HXMT 卫星在发射后首先进行大约三个月的在轨测试，随后展开科学观测，其中约 1 年时间用于巡天观测，其余时间用于定点和小天区观测。用于科学观测的在轨运行时间累积不少于 4 年。在卫星观测期间，根据科学任务需求，HXMT 卫星有巡天扫描、定点观测和小天区扫描共三种观测模式。通过卫星姿态的调整进行工作模式间的相互切换，从而实现巡天扫描、定点观测、小天区扫描、银道面扫描、定标观测等观测任务。在不同观测模式下，有效载荷本身的工作方式相同，即对入射 X 射线进行逐光子事例记录。

(2) 数据产品分级

HXMT 卫星数据检索发布系统中的数据特指科学数据，即由卫星原始数据衍生出的各级科学数据产品，包括事例数据、载荷工程数据、地面辅助数据等。本文主要针对“科学数据产品”进行定义、分级和对外发布。

遵循国际天文数据分级标准规范，同时综合考虑 HXMT 卫星数据产品的处理程度，对 HXMT 卫星科学数据产品进行分级，如表 3.1 所示。0 级数据产品是指原始观测数据。1 级是经数据组织、重构、换算等处理后得到的数据产品，包括定点或小天区观测产生的 1L 级数据，以及巡天观测阶段获得的 1S 级数据。其中，1L 级数据按曝光 ID 组织，每次曝光对应的数据内容包括所有可面向用户发布的事例数据、工程数据和地面辅助数据；1S 级数据内容包括巡天观测阶段获得的巡天观测数据（包括事例数据、工程数据等）、按天区重新组织的巡天事例数据、经分析处理后的巡天结果数据。2 级科学数据产品是对 1 级数据产品进行不可逆的标准分析处理后得到的具有科学意义的数据产品。

按照 HXMT 卫星数据政策的要求，仅将 1L、1S、2L 级科学数据产品面向科学用户开放公开发布，0 级科学数据产品均不对外公开发布。

表 3.1 HXMT 科学数据产品级别

数据级别		处理程度
标准科学 数据产品	0 级数据产品	经地面站接收、解调、帧同步、解码、解扰后生成的由卫星原始数据数据传输帧组成的数据文件。根据卫星下行信道的不同，分为数传原始数据和遥测原始数据两类。
	1 级	经过数据组织、重构、换算等处理过程得到的可供科学用户进行科学研究的观测数据产品。
	1 级提案观测数据 (1L) 1 级巡天观测数据 (1S)	
	2 级科学数据产品	按照目标天体的特性选择一组预先定义的标准参数，应用标准的数据分析流程对 1 级数据进行分析得到的具备科学意义的分析结果。

(3) 归档目录结构

在 HXMT 科学数据档案库根目录下，按照数据级别分别设置 3 个一级目录，分别存放 1L 级、1S 级和 2L 级数据产品，如图 3.2 所示。下面具体描述各级科学数据产品对应的归档目录结构。其中，1L 级数据以“提案-观测-曝光”为归档组织线索，1S 和 2L 级数据均按“时间”顺序归档组织。

```

SciArchive
|
---1L
---1S
---2L

```

图 3.2 顶级目录结构

① 1L 级数据产品目录结构

1L 级数据产品是面向科学用户正式发布的、由定点或小天区观测产生的提案数据产品，包括了每个提案观测时段内所有可公开的、具有科学意义的数据产品。

1L 数据以“提案-观测-曝光”作为数据组织的线索。每个“提案”通常会包含多次“观测”，每次“观测”以观测目标的切换或者仪器状态参数的改变为划分依据。每次“观测”又人为地按观测时段分割为若干次“曝光”，每次“曝光”长度选择在 3 小时左右。因此，以“提案-观测-曝光”为归档组织依据，能逐层递进地定位到所需数据产品。1L 级数据目录结构如图 3.3 所示。

```

|---1L
|   |---A01
|   |---A02
|   |---A03
|   |   |---P0103041
|   |   |   |---观测号
|   |   |   |   |---曝光ID
|   |   |   |   |---AUX（地面辅助数据）
|   |   |   |   |---ACS（姿态轨道数据）
|   |   |   |   |---CAL（标定信息）
|   |   |   |   |---CRAFT（平台工程数据）
|   |   |   |   |---HE（高能望远镜）
|   |   |   |   |---LE（低能望远镜）
|   |   |   |   |---LOG（日志数据）
|   |   |   |   |---ME（中能望远镜）
|   |   |---P0103052

```

图 3.3 HXMT 1L 级数据目录结构

1L 目录下按提案批次划分二级子目录，采用字母 A+两位数字的命名方式，数字从 01 开始，表示第 N 批提案。按提案号划分三级子目录，提案号是每一个获得批准的观测提案唯一的编号，由字母 P+2 位提案批次数+2 位表示提案类型的数字+3 位数字序号组成。按观测号划分四级子目录，观测号定义为提案号+2 位任务号+3 位观测序号。按曝光 ID 划分五级子目录，曝光 ID 由观测号+2 位曝光序号+8 位观测日期+2 位曝光开始时间在此次观测中的天序号+2 位该曝光在当天的序号组成。

HXMT 卫星数据以曝光 ID 作为数据产品的组织单位，每一个以曝光 ID 命名的子目录中存放有每次曝光过程中产生的全部观测数据产品。根据数据种类和来源又划分 8 个六级子目录，具体包括：AUX 目录用于存放与该 PI 有关的地面辅助数据，例如好时间段（GTI）、观测时间线、命令历史、数据质量报告等；ACS 目录用于存放姿控轨控数据，包括轨道数据、姿态数据、指向数据；CRAFT 目录用于存放卫星平台测量得到的各种数据；HE、LE、ME 目录分别用于存放 HE、LE、ME 望远镜探测和测量得到的各种事例、工程数据，也包括少量由卫星平台采集、与 HE、LE、ME 望远镜有关的数据；LOG 目录用于存放数据接收、处理过程中的日志信息；CAL 目录用于存放适用于此次曝光数据的能量响应矩阵文件。

②1S 级数据产品目录结构

1S 级数据产品是面向科学用户正式发布的、巡天观测获得的数据产品，包括巡天观测阶段可用于科学研究的数据产品，覆盖了除了 SAA 区（South Atlantic Anomaly，南大西洋异常区）、地球遮挡之外其余时段的科学观测数据。

1S 数据以时间为主要组织线索。1S 目录结构如图 3.4 所示。1S 目录下按年份划分二级子目录，采用字母 Y+4 位年份的方法命名；以天为单位划分三级子目录，采用 8 位数字日期+4 位天序号的规则命名；每个三级子目录下又根据数据种类和来源划分 8 个四级子目录，包括 AUX（地面辅助数据）、ACS（姿态轨道数据）、CAL（标定信息）、CRAFT（平台工程数据）、HE（高能望远镜）、LE（低能望远镜）、LOG（日志数据）、ME（中能望远镜），用于存放一天的巡天观测数据。

```

1S
|---Y2014
|---Y2015
|   |---20150803-0380
|   |---20140803-0380
|   |   |---AUX（地面辅助数据）
|   |   |---ACS（姿态轨道数据）
|   |   |---CAL（标定信息）
|   |   |---CRAFT（平台工程数据）
|   |   |---HE（高能望远镜）
|   |   |---LE（低能望远镜）
|   |   |---LOG（日志数据）
|   |   |---ME（中能望远镜）

```

图 3.4 HXMT 1S 级数据目录结构

③2L 级数据产品目录结构

2L 级数据产品是面向用户发布、经标准数据处理分析流程得到的科学数据产品。2L 数据以“提案-观测”作为数据组织线索。2L 目录结构如图 3.5 所示。在 2L 目录下，划分以提案年度命名的二级子目录，以提案号命名的三级子目录，以观测号命名的四级子目录。由于标准数据分析过程以一次观测的数据内容为处理单位，因此所有 2L 级数据产品实体文件直接存放在以观测号命名的四级子目录中。

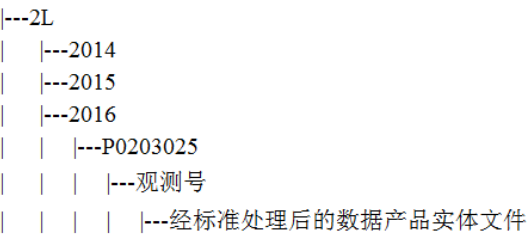


图 3.5 HXMT 2L 级数据产品目录结构

2、HXMT 与 Chandra 数据产品特点对比

因 HXMT 卫星尚未发射，各级数据产品尚未产生，数据产品的命名方式及文件内容尚未完全确定，故暂时需要采用其他天文项目的数据作为系统测试数据。经详细的调研分析，发现比较适合采用 Chandra 卫星的天文数据。

下面对 HXMT 卫星与 Chandra 卫星的数据产品特点进行分析与比对。

(1) 科学目标

HXMT 卫星的科学目标之一为高灵敏度宽波段 X 射线巡天，Chandra 卫星的科学目标为观测天体的 X 射线辐射。两者的科学目标均包括观测天体的 X 射线辐射。

(2) 数据分级

HXMT 卫星和 Chandra 卫星均依据数据处理程度进行严格的数据分级定义，数据级别均包含 L0 级、L1 级和 L2 级，但在具体划分细节方面还存在差异。具体数据分级异同点对比情况如表 3.2 所示。

表 3.2 HXMT 卫星和 Chandra 卫星数据分级对比

数据 级别	同	异	
		HXMT 卫星	Chandra 卫星
L0 级	基本未经处理的原始观测数据。	经地面站接收、解调、帧同步、解码、解扰后生成的由卫星原始数据数据传输帧组成的数据文件。根据卫星下行信道的不同，分为数传原始数据和遥测原始数据两类。	以 FITS 格式保存。
L1 级	去除仪器依赖，包括科学用户在常规的数据分析过程中需要用到	经过数据组织、重构、换算等处理过程得到的可供科学用	单次观测时间段内生成的、经过定标的数据产品，包括事例数据、星历数据、指向数据等。

	的事例、工程和地面辅助数据集。处理过程可逆，可对外公开发布。	户进行科学研究的观测数据产品。	
L1.5 级		无	经更高级定标处理后的数据产品，只应用于特定数据，如光栅数据和太阳系观测数据。
L2 级	在标准数据处理分析过程中均以一次观测的数据内容为处理单位，得到具备科学意义的分析结果，可对外发布。处理过程不可逆。	按照目标天体的特性选择一组预先定义的标准参数，应用标准的数据分析流程对 1 级数据进行分析得到的具备科学意义的分析结果。	对同一观测号内所有观测时间段数据进行合并、过滤处理后得到的数据产品，包括星源列表、L2 事例文件、光栅数据双色散光谱等。
L3 级		无	处理为更高级的聚合分析数据、星表数据，以便进行交叉证认等分析。

(3) 归档目录结构

均采用分级的目录组织形式进行数据归档与存储，但各子目录具体划分依据和命名规则各有不同。例如，HXMT 卫星 1L 级数据产品以“提案-观测-曝光”作为数据组织的线索，存放在 1L 目录下，划分以提案批次命名的二级子目录，以提案号命名的三级子目录，以观测号命名的四级子目录，以曝光 ID 命名的五级子目录。Chandra 根目录下主要分为 byobsid、science、cal、er 四个子目录，science 包含所有科学观测数据，cal 包含所有定标数据，er 包含所有工程数据，byobsid 包含所有通过观测号可直接访问获取的数据产品，可以是科学观测数据，也可以是定标数据，或工程数据，但不能同时兼有三种类型数据。byobsid 目录按观测号末位数字划分为 0~9 共 10 个子目录；每个子目录下面存储有独立的、以观测号为组织单位的数据产品实体文件。

(4) 数据组织单位

HXMT 卫星 1L 级数据以曝光 ID 为组织单位，1S 级数据以时间为组织单位，2L 级数据以观测号为组织单位；Chandra 卫星各级数据产品都以观测号作为数据组织单位。由此可见，只有 HXMT 卫星的 2L 级数据与 Chandra 卫星数据的组织单位相同，将针对某次观测的所有相关数据产品实体文件直接存放在以观测号命名的文件夹中。

(5) 数据产品种类

HXMT 卫星 1L 级和 1S 级科学数据产品根据数据种类和来源划分为 HE（高能望远镜）、LE（低能望远镜）、ME（中能望远镜）、AUX（地面辅助数据）、ACS（姿态轨道数据）、CAL（标定信息）、CRAFT（平台工程数据）、LOG（日志数据）8 大类，共约 30 多种文件。具体存放数据产品实体的文件类型如表 3.3 所示。

表 3.3 HXMT 卫星数据产品文件类型列表

文件 ID	数据名称	数据名称	文件类型缩写
HE	HE 事例数据	HE_Events	HE-Evt
	HE 科学事例数据	HE_Scientific_Events	HE-SciEvt
	HE 标定源事例数据	HE_Radioactive_Events	HE-RadEvt
	HE 盲探测器事例数据	HE_Sheltered_Detector_Events	HE-ShltEvt
	HE 设备状态数据	HE_Instrument_Status	HE-InsStat
	HE 温度监测数据	HE_Temperature	HE-TH
	HE 计数类数据	HE_Counts	HE-Cnts
	HE 死时间计数	HE_DeadTime	HE-DTime
	HE 高压监测数据	HE_High_Voltage	HE-HV
	HE 粒子监测器数据	HE_Particle_Monitor	HE-PM
	HE 探测器阈值	HE_Threshold	HE-Thresh
ME	ME 事例数据	ME_Events	ME-Evt
	ME 科学事例数据	ME_Scientific_Events	ME-SciEvt
	ME 标定源事例数据	ME_Radioactive_Events	ME-RadEvt
	ME 盲探测器事例数据	ME_Sheltered_Detector_Events	ME-ShltEvt
	ME 电路监测数据	ME_Circuit_Parameter	ME-CirPara
	ME 设备状态数据	ME_Instrument_Status	ME-InsStat
	ME 温度监测数据	ME_Temperature	ME-TH
	ME 计数类数据	ME_Counts	ME-Cnts
LE	LE 事例数据	LE_Events	LE-Evt
	LE 科学事例数据	LE_Scientific_Events	LE-SciEvt
	LE 标定源事例数据	LE_Radioactive_Events	LE-RadEvt
	LE 盲探测器事例数据	LE_Sheltered_Detector_Events	LE-ShltEvt
	LE 随机触发事例数据	LE_Rand_Sample_Events	LE-RandEvt
	LE 电路监测数据	LE_Circuit_Parameter	LE-CirPara
	LE 设备状态数据	LE_Instrument_Status	LE-InsStat
	LE 温度监测数据	LE_Temperature	LE-TH
	LE 计数类数据	LE_Counts	LE-Cnts
	LE 阈值	LE_Detector_Threshold	LE-Thresh
ACS	姿态数据	Attitude	Att

	轨道数据	Orbit	Orbit
CRAFT	有效载荷 TH 数据	TH_of_Payload_Cabin	TH-PL-Cabin
AUX	事例过滤文件	Event_Filter	Evt-Filter
	好时间段文件	Good_Time	GTI
	数据质量报告	Data_Quality_Report	Data-QA
	命令历史文件	Command_History	Comm-Hist
LOG	时间源包信息		
	数据产品生成日志		
	数据归档日志		
	标准数据分析日志		
	数据发布日志		

Chandra 卫星数据产品是以观测号为组织单位进行归档存储的，以观测号命名的文件夹下包含观测索引文件(oif.fits)、校验报告(V&VReport.pdf)，以及 primary 目录和 secondary 目录。primary 目录和 secondary 目录中具体包括全天区图像(full_img2.fits)、天区中央高分辨率图像(cntr_img2.fits)、L2 级事例文件(evt2.fits)、坏点像素文件(bpix1.fits)、视场文件(fov1.fits)、轨道星历文件(orbit_XXX_eph1.fits)、L1 级事例文件(evt1.fits)、好时间段文件(flt1.fits)、姿态指向文件(pcadfXXX_asol1.fits/pcadfXXX_osol1.fits)、Mask 文件(msk1.fits)、观测时间线文件(mtl1.fits)、曝光统计文件(stat1.fits)、偏离值文件(bias0.fits)、参数块文件(pbk0.fits)、扫描质量文件(pcadfXXX_aqual1.fits)共 15 种数据产品文件。

(6) 数据存储格式

均主要采用 FITS 格式保存数据产品实体文件，此外还存有 jpeg 格式图片和 pdf 格式文档。

根据上述分析得知，HXMT 卫星与 Chandra 卫星在归档目录结构、数据组织单位方面存在较大差异，故数据模型、数据库和文件系统均严格针对 HXMT 卫星进行设计。又因两颗卫星的科学目标一致，探测范围相似，数据分级标准、数据存储格式、数据产品种类方面也有重合之处，HXMT 卫星数据产品文件内容又尚未确定，故采用 Chandra 数据产品实体文件作为系统底层测试数据。研制的原型系统若能在 Chandra 卫星数据上运行成功，则会为今后应用于真正的 HXMT 卫星数据奠定良好的基础。

3.4.2 分层数据模型设计

为了方便 HXMT 卫星数据的对外发布,并有效解决数据资源的互操作问题,设计了分层数据模型,如表 3.4 所示。

表 3.4 分层数据模型

	存储格式	存储位置	数据内容
互访问数据	VOTable	无	元数据信息
元数据	数据库表	数据库	目标名称、赤经、赤纬、观测号、提案号、载荷名称、观测模式、观测起始时间、有效曝光时间、数据发布状态、数据发布时间、文件路径
数据实体文件	FITS、jpg、pdf	文件系统	HE/ME/LE 事例数据、HE/ME/LE 科学事例数据、HE/ME/LE 标定源事例数据、HE/ME/LE 设备状态数据、HE/ME/LE 计数类数据、姿态数据、轨道数据、有效载荷 TH 数据、事例过滤文件、好时间段文件、数据质量报告、数据发布日志等

1、数据实体文件

最底层为数据实体文件,以 HXMT 卫星各级数据产品归档目录结构保存在文件系统中,包含全部的 HXMT 卫星数据产品,如 HE/ME/LE 事例数据、HE/ME/LE 计数类数据、姿态数据、轨道数据、事例过滤文件、好时间段文件、数据质量报告、数据发布日志等。存储格式较为单一固定,处理后的科学观测图像分别以 jpg 和 FITS 格式各存储一份,数据质量报告、各类日志文件等文档资料以 pdf 格式存储,其他科学数据产品均以标准的 FITS 格式存储。

2、元数据

元数据是对数据实体的特征属性提供结构化描述的数据。提取元数据信息充当数据实体文件的索引,便于对数据量大且种类复杂的数据实体文件进行高效管理、检索与发布。本文从数据实体文件中提取出关键的元数据信息导入到关系型数据库中,以数据库表的形式存储元数据。元数据是对数据实体文件的精炼,具体包含从姿态轨道数据 FITS 文件头中提取的赤经、赤纬信息,从 1L 级数据产品文件名中提取的提案号、观测号、曝光 ID、观测日期、数据文件类型、载荷类型、观测模式、数据级别,归档过程中添加的数据发布时间、数据发布状态、以及文件具体存储路径信息。

3、互访问数据

为有效融入VO环境，方便不同天文数据中心进行数据互访问、互操作，采用虚拟天文台中通用的数据互操作标准VOTable格式进行数据封装，来传递不同系统间的天文数据。VOTable支持TABLEDATA、FITS、BINARY三种形式的数
据。本文的VOTable采用TABLEDATA形式传递查询结果数据。VOTable设计如图3.6所示，主要可划分为元数据描述和查询结果数据两大部分。元数据部分是
与数据库元数据表中各字段相对应的属性描述数据，封装于VOTable的<FIELD>
中，包括

obsid,ra,dec,target_name,instrument,proposal_num,observation_mode,observation_start_time,exposure_time,publishing_status,publishing_date（观测号、赤经、赤纬、
目标名称、载荷类型、提案号、观测模式、观测起始时间、有效曝光时间、数据
发布状态、数据发布时间），对应于数据库表中

OBS_ID,RA_NOM,DEC_NOM,OBJECT,INSTRUME,PROPOSAL,OBS_MODE,DATE_OBS,EXPOSURE,STATUS,DATE字段；查询结果数据部分是指检索数据库
后将查询结果数据反馈回来，封装于VOTable的<DATA>中的数据，呈表列显示。

HXMT卫星数据的VOTable设计实例如图3.6所示。

```

<?xml version="1.0" encoding="utf-8"?>
<!--This VOTable was generated by the HXMT Data Retrieval and Publishing System.
We make every attempt to accurately reproduce the original content.-->
<VOTABLE version="1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.ivoa.net/xml/VOTable/v1.2" xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2
http://www.ivoa.net/xml/VOTable/v1.2" xmlns:stc="http://www.ivoa.net/xml/STC/v1.30">
<RESOURCE type="results">
<INFO name="QUERY_STATUS" value="OK" />
<PARAM datatype="char" name="CoordRefFrame" ucd="VOX:STC_CoordRefFrame" value="ICRS" arraysize="*" />
<PARAM datatype="char" name="CoordProjection" ucd="VOX:WCS_CoordProjection" value="TAN" arraysize="*" />
<PARAM datatype="char" name="Bandpass" ucd="VOX:BandPass_ID" value="X-ray" arraysize="*" />
<PARAM datatype="char" name="BandpassUnit" ucd="VOX:BandPass_Unit" value="keV" arraysize="*" />
<TABLE name="HXMT Data">
<DESCRIPTION>Simple Cone Search for HXMT Data
Explanations and Statistics of UCDS: http://www.ivoa.net/documents/REC/UCD/UCDlist-20070402.htm
</DESCRIPTION>
<FIELD datatype="int" name="obsid" ucd="meta.id">
<DESCRIPTION>HXMT observation identifier</DESCRIPTION>
<VALUES null="-2147483648" />
</FIELD>
<FIELD datatype="double" name="ra" ucd="pos.eq.ra" unit="deg">
<DESCRIPTION>Right ascension, ICRS (J2000)</DESCRIPTION>
</FIELD>
<FIELD datatype="double" name="dec" ucd="pos.eq.dec" unit="deg">
<DESCRIPTION>Declination, ICRS (J2000)</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="target_name" ucd="meta.id,src">
<DESCRIPTION>Target name</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="instrument" ucd="instr">
<DESCRIPTION>Instrument name</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="proposal_num" ucd="obs.proposal">
<DESCRIPTION>Proposal number, including several related observations</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="observation_mode" ucd="obs.param">
<DESCRIPTION>Observation mode, including Pointing, Survey and Area</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="observation_start_time" ucd="time.start,obs">
<DESCRIPTION>Start time of observation</DESCRIPTION>
</FIELD>
<FIELD datatype="double" name="exposure_time" ucd="time.duration,obs.exposure" unit="ks">
<DESCRIPTION>Total exposure time</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="publishing_status" ucd="meta.code,status">
<DESCRIPTION>Publishing status: unpublished, published</DESCRIPTION>
</FIELD>
<FIELD arraysize="*" datatype="char" name="publishing_date" ucd="time.release" xtype="adql:TIMESTAMP">
<DESCRIPTION>Date/time when data is available to the public</DESCRIPTION>
</FIELD>
<DATA>
<TABLEDATA>
<TR>
<TD>934</TD>
<TD>210.79375</TD>
<TD>54.359667</TD>
<TD>M101</TD>
<TD>ACIS</TD>
<TD>05600587</TD>
<TD>POINTING</TD>
<TD>2004-03-07T09:29:06</TD>
<TD>52093.500</TD>
<TD>published</TD>
<TD>2012-11-12T21:35:31</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
<INFO name="QUERY_STATUS" value="OK" />
</RESOURCE>
</VOTABLE>

```

图 3.6 VOTable 设计

3.4.3 数据库设计

因对外公开发布的数据产品中，1S 和 2L 级数据产品尚未定义完全，故以

1L 级数据产品为例进行数据库设计，其余数据产品不多做赘述。

1、概念设计

综合归纳和抽象系统需求，进行数据库概念设计，形成 E-R 图，如图 3.7 所示。

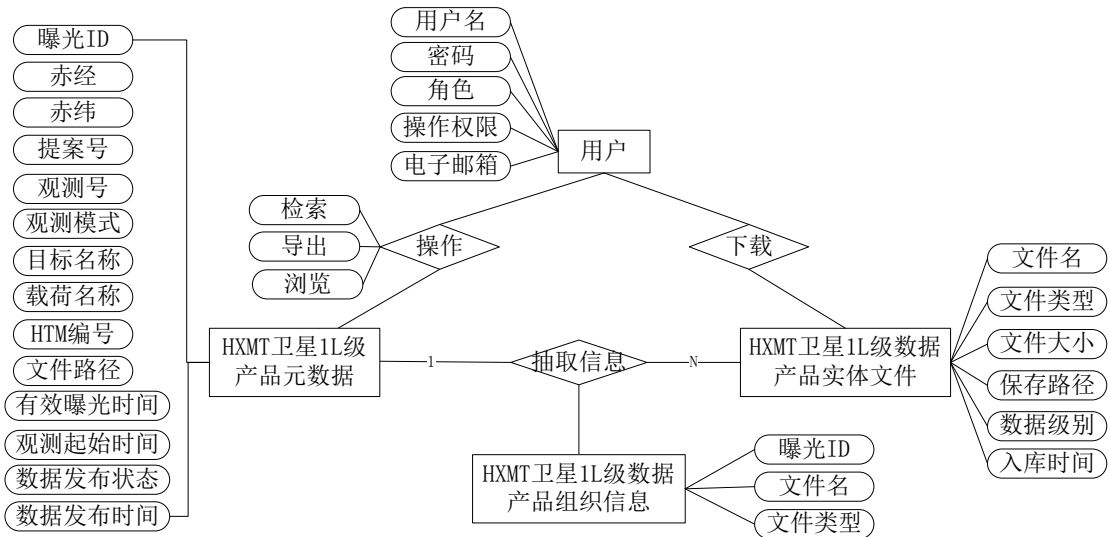


图 3.7 E-R 图

2、逻辑设计

设计数据库模式，并以表的形式展示出来。

建立“HXMT 卫星 1L 级数据产品实体文件信息表”，用于存放每个数据产品实体文件的相关文件信息，如表 3.5 所示。建立“HXMT 卫星 1L 级产品元数据表”，用于存放从数据产品实体文件的 FITS 头和文件名中抽取的元数据信息，如表 3.6 所示。经分析可知，以曝光 ID 为组织单元的文件夹下存储有与本次曝光相关的全部 1L 级数据产品实体文件，具体包括 HE、LE、ME、AUX、ACS、CAL、CRAFT、LOG 八大类共约 30 多种数据文件。为体现这种元数据与数据产品实体文件之间一对多的关系，建立“HXMT 卫星 1L 级数据产品组织信息表”，将实体文件信息表和元数据表联系起来，设计如表 3.7 所示。“用户信息表”设计如表 3.8 所示。

表 3.5 HXMT 卫星 1L 级数据产品实体文件信息表

表名	1L_File_Info			
列名	数据类型	主/外键	空/非空	注释

FILE_NAME	VARCHAR(100)	Primary	Not Null	文件名
FILE_TYPE	VARCHAR(30)		Not Null	文件类型缩写(HE-Evt,Att,Orbit 等)
FILE_SIZE	TEXT		Not Null	文件大小(单位:K)
PATH	TEXT			文件保存路径
LEVEL	VARCHAR(10)			数据级别(1L/1S/2L)
IMPORT_TIME	TIMESTAMP(14)			入库时间(YYYYMMDDHHMMSS)

表 3.6 HXMT 卫星 1L 级产品元数据表

表名	1L_Product_Meta			
列名	数据类型	主/外键	空/非空	注释
EXP_ID	VARCHAR(50)	Primary	Not Null	曝光 ID
RA_NOM	TEXT		Not Null	赤经(deg)
DEC_NOM	TEXT		Not Null	赤纬 (deg)
OBJECT	VARCHAR(10)			目标名称
OBS_MODE	VARCHAR(5)			观测模式(P/R, 代表定点/小天区)
INSTRUME	VARCHAR(5)			载荷名称(HE/ME/LE)
PROPOSAL	VARCHAR(50)			提案号
OBS_ID	VARCHAR(50)			观测号
EXPOSURE	TEXT			有效曝光时间(s)
DATE_OBS	TEXT			观测起始时间
DATE	TEXT			数据发布时间
STATUS	VARCHAR(10)			数据发布状态(published/unpublished)
PATH	TEXT			产品保存路径
HTM_NAME	VARCHAR(100)			HTM 索引编码值

表 3.7 HXMT 卫星 1L 级数据产品组织信息表

表名	1L_Product_Org			
列名	数据类型	主/外键	空/非空	注释
EXP_ID	VARCHAR(50)	Primary/Foreign	Not Null	曝光 ID
FILE_NAME	VARCHAR(100)	Primary/Foreign	Not Null	文件名
FILE_TYPE	VARCHAR(30)		Not Null	文件类型缩写(HE-Evt,Att,Orbit 等)

表 3.8 用户信息表

表名	User_Info			
列名	数据类型	主/外键	空/非空	注释
USER_NAME	VARCHAR(20)	Primary	Not Null	用户名
PASSWORD	VARCHAR(20)		Not Null	密码
EMAIL	VARCHAR(20)			邮箱
ROLE	VARCHAR(20)		Not Null	角色
RIGHT	VARCHAR(20)		Not Null	操作权限

3、物理设计

在 MySQL5.5 数据库中进行物理数据库结构设计，物理设计模型图如图 3.8 所示。数据库索引设计请见 3.4.4 节。

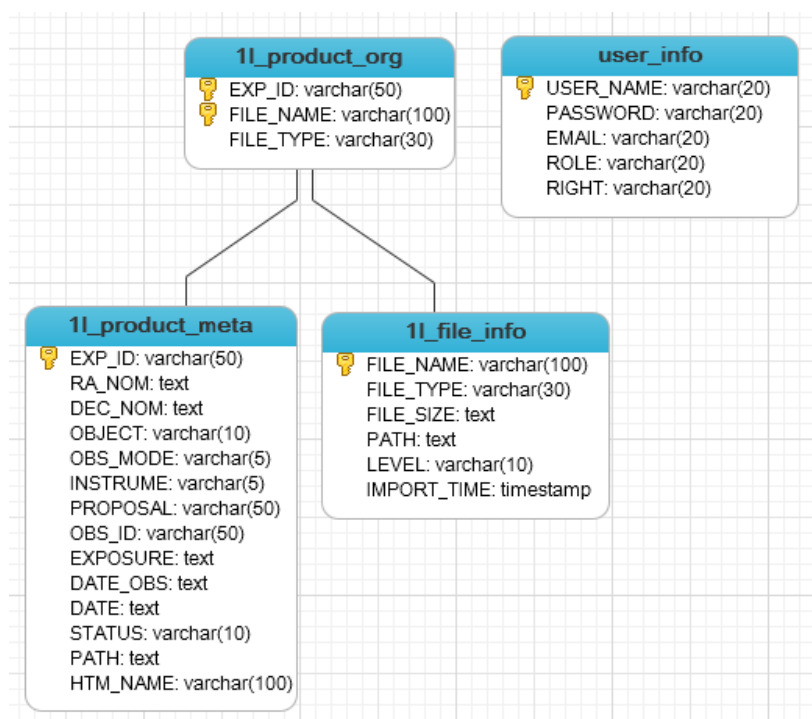


图 3.8 数据库物理设计模型图

3.4.4 数据库索引设计

1、数据库索引方式对比

在访问海量天文数据的过程中，建立数据库索引是提高其检索浏览效率必不可少的重要途径。数据库索引可分为唯一索引、普通索引、主键索引、聚集索引四种类型。原型系统在执行锥形检索功能时，需对用户输入的赤经、赤纬坐标信息进行角距离计算。锥形检索中计算角距离时，需应用到球面坐标系中球面两点

(α_0, δ_0) 和 (α_1, δ_1) 间的大圆距离公式： $r = \cos^{-1}[\sin(\delta_1) * \sin(\delta_0) + \cos(\delta_1) * \cos(\delta_0) * \cos(\alpha_1 - \alpha_0)]$ 。具体的 SQL 语句为"SELECT * FROM 1l_product_meta WHERE DEGREES(ACOS(SIN(RADIANS(DEC_NOM))* SIN(RADIANS("+dec+")) + COS(RADIANS(DEC_NOM)) * COS(RADIANS("+dec+")) * COS(RADIANS(RA_NOM-"+ra+")))) <"+sr+";"。由此可见 Where 子句中需对赤经、赤纬(RA_NOM, DEC_NOM)字段进行三角函数操作，这将导致数据库引擎放弃使用索引而进行全表扫描^[39]。对任何一个数据库管理系统而言，多次顺序进行三角函数计算，将大大降低数据库的检索性能。因此，针对此类问题需要设计一种特殊的索引方式来提升锥形检索效率。

目前，针对锥形检索，天文学界主要有三种建立索引方式。

(1) 空间一维索引

比较典型的空间一维索引当属 B-tree 索引。对赤经 (RA) 和赤纬 (DEC) 分别建单列一维 B-tree 索引，或对 (RA, DEC) 两列一起建多列一维 B-tree 索引。该方法可先检索出满足一定范围内的数据子集，例如基于 DEC 的一维索引可快速定位球面上一个维度跨度上的一圈数据，该数据子集通常远远大于检索条件确定的一小片空间区域。然后对数据子集仍要进行球面距离计算。由此可见，一维 B-tree 索引方法在缩小检索范围的情况下可部分提升检索效率，但仍需对较大范围的数据子集进行球面计算与顺序扫描，在数据规模较大的情况下，其性能提升空间有限。

(2) 空间二维索引

空间二维索引包括 R-tree、SR-tree、G-tree 等索引，其中较为成熟的当属 R-tree 二维索引。目前，部分数据库软件已经支持 R-tree 空间索引的建立，例如 MySQL、PostgreSQL 等。但 R-tree 索引技术尚缺乏配套的并发控制机制来确保数据一致性，而且无法保证在索引建立、数据增删改等各方面的综合性能优越性^[40]。例如，Kalpakis^[41]曾在文章中指出，在 Informix 数据库中为 USNO-A1.0 星表建立 R-tree 二维索引竟然费时 25 天。Clive Page^[42]针对空间二维索引进行性能测试实验，测试结果同样显示数据导入和索引建立过程会耗费较长的时间，且索引占用较大的存储空间，PostgreSQL 数据库中建立 R-Tree 索引后的数据表大小

约是原来的 12 倍。从技术成熟度、索引建立效率、存储空间利用率等各方面综合来看，空间二维索引技术在天文学界的应用范围十分有限。

(3) 伪空间索引

伪空间索引是指利用某种映射关系将二维的空间参量映射到一维参数空间，把 (RA, DEC) 空间参数合二为一，从而降低检索维度^[23]。目前，天文学界通常采用伪空间索引方式来提升检索效率。

伪空间索引的具体建立方法分为多级三角划分法 (Hierarchical Triangular Mesh, 简称 HTM) 和多级等面积同纬度划分法 (Hierarchical Equal Area isoLatitude Pixelisation, 简称 HEALPix) 两种方式。两种算法的基本思路一致，均采用迭代方式对整个天区进行划分，然后按照一定的编码规则为每个天区单元编号，该编号与 (RA, DEC) 二维参数产生了一定的映射关系，从而实现了从二维空间到一维空间的唯一映射。大多成熟的数据库系统均可采用一维 B-tree 索引技术对天区单元编号建立索引并实施操作，进而实现快速检索的目的。建立伪空间索引的实质就是通过对天区进行球面分割来过滤掉大量不相关数据，不断缩小检索范围，以便快速高效地定位数据，提高检索访问性能。

HEALPix 已被 CMB 卫星、COBE 卫星、IRAS 卫星、Planck 卫星等卫星项目采用。HTM 也已经应用于 SDSS 星表、欧洲空间局 (ESA) GAIA 等项目的数据检索发布过程中。考虑到 HTM 算法提供相应的 Java 开发包，而系统研制过程中采用了 Java 编程语言，故本文选用 HTM 索引算法，来对存储于数据库中的天文数据建立索引。

2、HTM 索引方法介绍

HTM 索引算法中划分方式首先将整个天区划分为 8 等份，上下各 4 个球面三角形，然后以每个球面三角形各边中点为新顶点进行四等分，依此类推。划分级次为 0 到 5 级的划分结果情形如图 3.9 所示。不同划分级次所对应的单元数和单元面积也各有不同，级次越高，单元数越多，单元面积也就越小，如图 3.10 所示。HTM 的编码方式如图 3.11(a)(b)所示。HTM 索引采用四叉树结构存储，处于根节点位置的最顶层 8 块球面三角形依次命名为 N0、N1、N2、N3、S0、S1、S2、S3，每个球面三角形又划分为四个小三角形，依次按 0~3 命名。例如，

某节点名称为 N201301，其中，N2 代表根节点名称，后接 5 位数字代表采用 5 级划分层次，5 位编码唯一区分某个球面三角形^[43]。

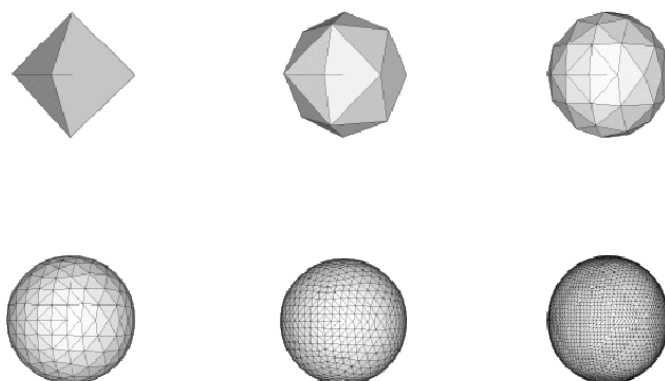


图 3.9 HTM 天区划分方法

级次	面积 (arcsec ²)	单元数
10	1.77E1	8,388,608
11	4.13E0	33,554,432
12	1.11E0	134,217,728
13	2.77E-1	536,870,912
14	6.92E-2	2,147,483,648
15	1.73E-2	8,589,934,592
20	1.69E-5	8,796,093,022,208
25	1.65E-8	9,007,199,254,740,922

图 3.10 HTM 不同划分级次对应的单元数和单元大小

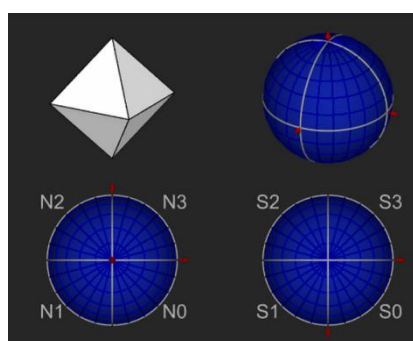


图 3.11 (a) HTM 编码方法

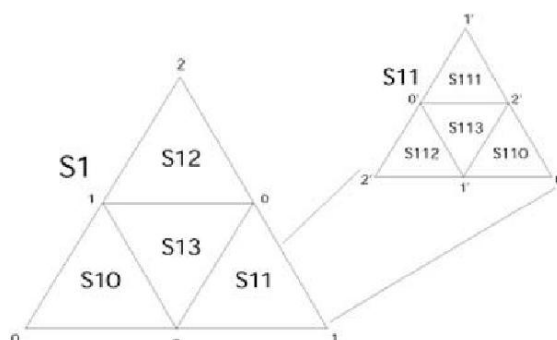


图 3.11 (b) HTM 编码方法

本文引入 HTM 索引，总体设计思路是：在数据库中对包含赤经、赤纬信息的元数据表（即 1l_product_meta 表）增加一列 HTM_NAME 用于存储对应的天区单元编号，并对 HTM_NAME 列建立 B-tree 普通索引；当进行锥形检索时，

计算 (RA, DEC, SR) 三个参数形成的天区范围内所包含天区单元编号的集合, 提取出数据库中 HTM_NAME 在该集合内的数据子集, 再对数据子集中的每条数据进行球面距离计算, 检索出最终符合锥形检索条件的数据。该方法是典型的以空间换时间的数据库优化方式。针对每条数据记录计算相应的天区单元编号值, 作为 HTM_NAME 属性添加到数据库表中, 需要耗费一定的时间和数据库存储空间, 并增加工作量, 但考虑到本次修改操作是在建库过程中一次性完成的, 以后无需重复, 而后续的锥形检索是用户经常进行的操作, 因此以牺牲可接受范围内的存储空间提升锥形检索效率的做法是合理有效的。HTM 索引方法的精华在于先缩小查询数据子集范围再进行复杂的球面距离计算, 进而提升锥形检索服务的数据查找访问效率。

3.5 本章小结

本章首先在分析系统需求及设计目标的基础之上, 确定了检索浏览、数据下载、可视交互三大核心功能模块。其次, 对 IVOA 概念体系结构进行适当剪裁, 构建了一套符合自身需求的体系架构。然后, 重点研究了 HXMT 卫星数据产品的分级定义、归档目录结构以及与 Chandra 数据产品的比对情况, 并设计了数据实体文件、元数据信息、互访数据分离的分层数据模型来管理和使用数据。最后, 详细阐明了数据库的概念设计、逻辑设计与物理设计, 论证了采用 HTM 索引方法的可行性。接下来将详细阐述系统的具体实现步骤与运行结果情况。

第四章 系统实现与运行

4.1 技术开发框架

4.1.1 开发环境

编程语言选用 Java。Java 在封装性、扩展性和多线程方面都有良好的支持，同时能在不同软硬件平台上进行移植，具有良好的跨平台部署能力，符合系统研制需求。

开发环境选用 Eclipse IDE for Java EE Developers。Eclipse 提供了对 J2EE 应用环境的全面支持。

Web 服务的发布引擎选用 Apache Tomcat。Tomcat 是运行 Servlet、JSP、Java 的 Web 应用软件容器，轻量小型，且免费开源，支持最新 Web 标准，具有跨平台特性。故采用 Tomcat 来发布 Web 服务。

选用 CXF 框架开发 SOAP 样式 Web 服务。CXF 框架继承了 Celtix 和 XFire 两大开源项目的精华，提供了对 JAX-WS 的全面支持，并且默认集成了 Spring 框架^[44]。因本文采用 Java 语言并引入 Spring 框架进行系统研制，故选用可与 Spring 框架无缝融合的 CXF 框架进行 Web 服务开发，方便后续系统功能的集成。

数据库选用 MySQL。MySQL 关系型数据库具有体积小、速度快、成本低的特点，并且开放源码。

4.1.2 技术架构

1、MVC 模式

为实现各程序模块间的高内聚、松耦合，增强代码可读性、复用性，方便后续维护，故引入 MVC（Model-View-Controller）模式，将模型、视图、控制器分别开来进行独立开发。相应地，选择较为成熟的 Spring 框架来具体实现 MVC 模式。

MVC 模式通常分为表示层、控制层、业务逻辑层、DAO 层（Data Access Object Layer，又称为数据访问层）和数据存储层五层^[45]。与系统体系架构中的各分层可相互对应，其中，表示层即本系统的用户层，控制层、业务逻辑层和 DAO 层

对应于本系统的服务层，数据存储层包含在本系统的资源层中。

下面按本系统的体系架构，逐一描述各层应用到的技术，以及具体实现类。系统技术架构如图 4.1 所示。系统类图如图 4.2 所示。

用户层主要由 JSP（Java Server Pages）页面构成。

服务层具体细分为控制层、业务逻辑层、DAO 层。

（1）控制层主要由 controller 包实现，controller 包中含有 SearchController 类和 FileDownloadController 类。FileDownloadController 类用于响应用户下载请求，调用 ProductDAO 类查询数据库并获取文件系统内的数据产品实体文件，打包压缩后返回给用户；SearchController 类用于响应用户检索请求，提取检索条件，调用 ProductDAO 类检索数据库并返回检索结果，同时也响应用户 Aladin 关联检索、点源标注、可视浏览的请求，并转到相应的界面显示可视交互结果。

（2）业务逻辑层由 entity 模型包实现，entity 包中含有 Product 类。Product 类作为 JavaBean，具有观测号、提案号、载荷名称、观测模式、观测起始时间、有效曝光时间、数据发布状态、数据发布时间等成员属性，以及用于读写 1L 级数据产品元数据信息的成员函数。

（3）数据访问层（DAO 层）由 dao 包实现，dao 包含有 ProductDAO 类。ProductDAO 类作为业务逻辑层与数据库之间的中间件，用于封装各种数据库操作，并返回处理结果数据。除此之外，为方便代码重用，抽取部分小的功能点放入 util 实用包中，util 包含有读取 properties 配置文件的 ConfigUtil 类、负责打包压缩数据的 ZipUtil 类和负责锥形检索的 ConeSearch 类。

资源层中，将 HXMT 卫星各级科学数据产品存储在文件系统中；从 FITS 格式数据实体文件中提取的观测号、赤经、赤纬、提案号、载荷名称、观测模式、观测起始时间、有效曝光时间、数据发布状态、数据发布时间等数据信息，附加文件路径信息，作为元数据存储到 MySQL 数据库中，以方便上层数据访问与获取。

用户层	V (View) 系统界面负责用户请求的收集和反馈	视图层 JSP	
服务层	C (Controller) 控制器负责控制业务逻辑跳转	控制层	Spring 整合
	M (Model) 实现业务逻辑	业务逻辑层 JavaBean	
	DAO 负责实体关系映射，以及数据访问对象与数据存储层的交互	DAO层	
资源层	数据库+ 数据产品实体文件	数据存储层 MySQL数据库+文件系统	

图 4.1 系统技术架构

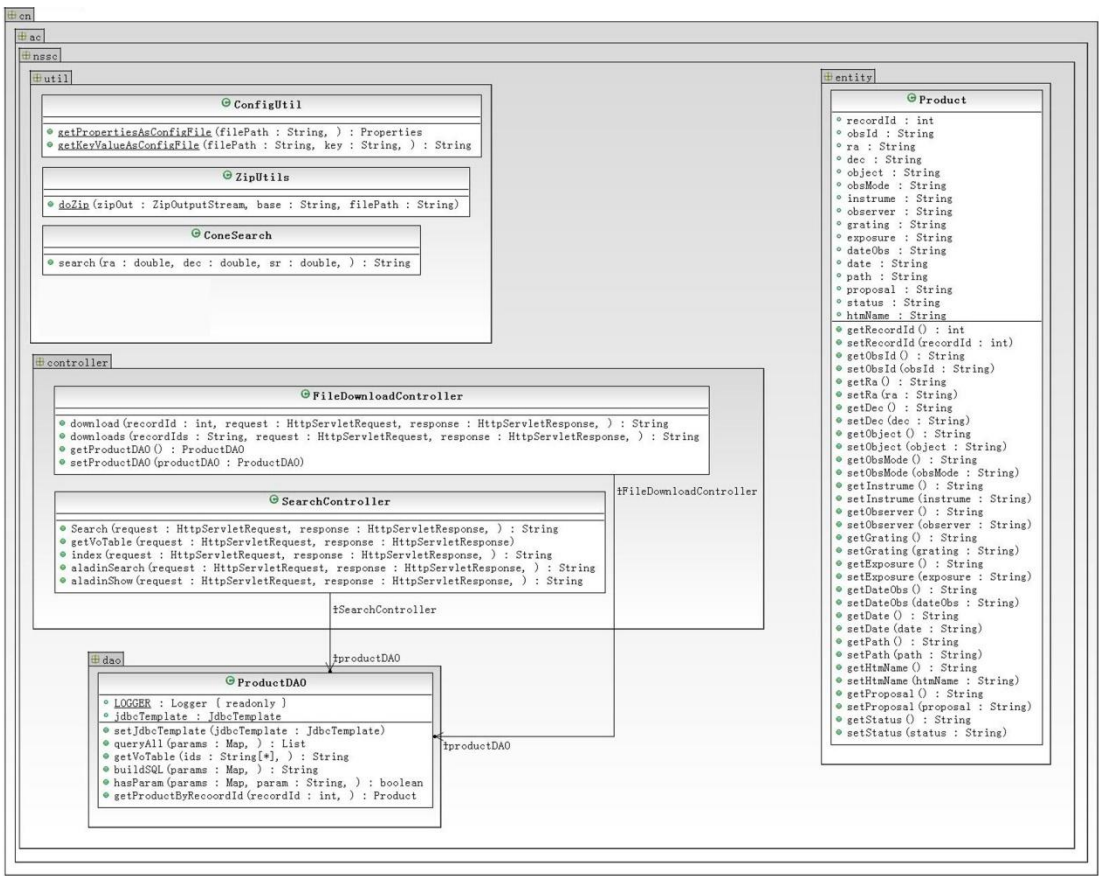


图 4.2 系统类图

2、Web Service 技术

本文主要采用 Web Service 来开发并发布符合 IVOA 标准规范的锥形检索服务。IVOA 中的 Web 服务并不完全符合严格意义上的 Web Service 标准规范。W3C 组织对 WebService 的定义主要强调了构建于 SOAP、WSDL、UDDI 之上的 Web 协议栈。然而，IVOA 不强制采用 SOAP 作为传输协议，不规定使用 WSDL 进

行服务描述，服务注册方面也有 VO 注册标准协议，与商业化的 UDDI 注册中心相比更为简单且具有针对性^[46]。考虑到 IVOA 中大量的 Web 服务是以 HTTP 作为数据传输协议，并且通常是由规范统一的 XML 格式作为消息内容，故在遵循 IVOA 标准协议的前提之上，采用 W3C 的 WebService 技术开发 Web 服务，以增强 Web 服务的普适性与扩展性，但在服务注册方面则采用 VO 自有的注册系统进行注册。

Web Service 开发过程中拥有 SOAP 样式和 REST (RepresentationalStateTransfer, 表征状态转移) 样式两种开发方案。SOAP 是一种轻量的、基于 XML 的标准协议，用于在 WEB 上交换结构化信息^[47]。开发的 SOAP 样式 Web 服务通常是面向活动的。而 REST 样式 Web 服务是面向资源的，主要通过 URI (Universal Resource Identifier, 统一资源标识符) 来识别和定位数据资源，并且针对这些资源而执行的操作全部是通过 HTTP 规范定义的 GET、PUT、POST、DELETE 来实现的^[23]。REST 开发简单，易于理解，但基于 REST 风格的 Web 服务对于语义的支持仅限于 URI，缺少足够的语义描述，缺少对传输协议和服务的定义，从而不利于在软件系统或应用服务中实现自动识别与调用。与之相比，SOAP 是一个工业标准，具备良好定义的协议与规则，通用范围较广，且 IVOA 的 WS 概要中明确提出建议采用 SOAP 标准^[48]。

综上，本文采用 SOAP 样式开发 Web 服务。选用同时支持 SOAP 和 REST 样式 Web 服务开发且能与 Spring 框架无缝融合的 CXF 框架，从而使得开发后的 Web 服务既符合 SCS 标准协议，又能被其他 Web 服务所调用，同时，还能方便地集成到 Spring 框架开发的系统中。

4.2 主要功能实现

4.2.1 检索浏览

1、流程图

检索浏览功能模块流程图如图 4.3 所示。用户在 HXMT 门户网站的数据检索界面中输入检索信息，系统验证检索信息的有效性。若有效，则根据检索信息判断所属的检索方式。(1) 若组合检索条件不为空，则提取不为空的组合检索条

件参数；若组合检索条件为空，则忽略组合检索中所有条件。(2) 若锥形检索条件不为空，则判断目标名称是否为空，若不为空，则调用名称解析服务将其解析为空间位置坐标（赤经、赤纬），若为空则直接提取赤经、赤纬、检索范围三项锥形检索条件参数。若锥形检索条件为空，则忽略锥形检索中所有条件。将两种检索方式中获取的检索请求参数拼接为 SQL 语句。检索数据库并返回 VOTable 格式的检索结果。对 VOTable 数据进行解析后，将检索结果显示于 Web 界面中以供用户浏览。

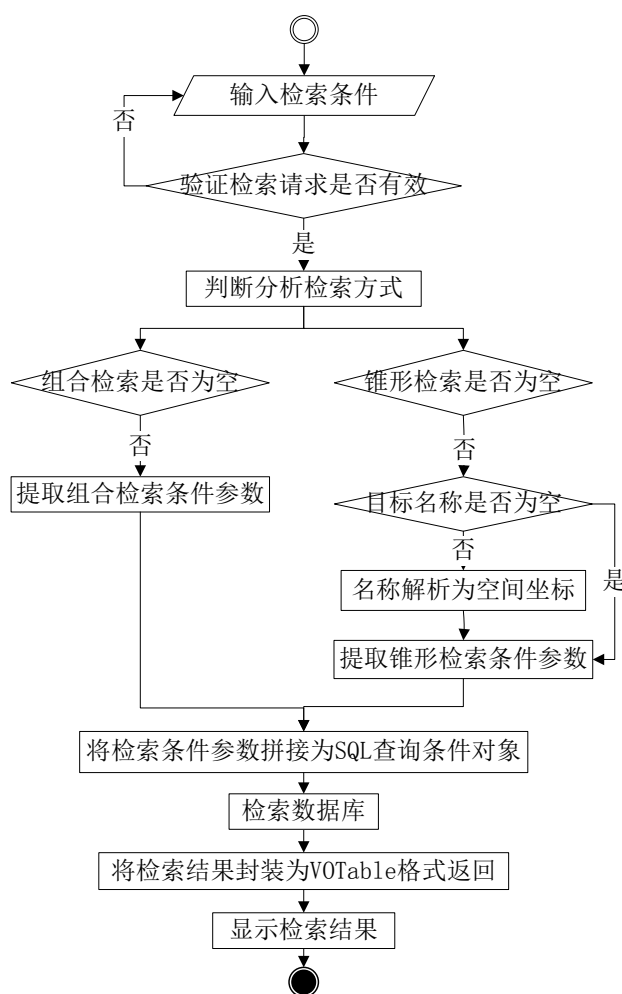


图 4.3 检索浏览功能模块流程图

2、Web Service 实现

根据 IVOA 制定的 SCS 协议，必须采用 Web 服务方式开发锥形检索服务。下面详述符合 SCS 协议、SOAP 样式的锥形检索服务的具体开发过程。SCS 服务必须遵循以下约束：服务必须接收并响应以标准 URL 方式传输的 HTTP GET 请求，标准 URL 具体格式为 `http://服务部署的域名/本地路径? 赤经 RA&赤纬`

DEC&检索半径 SR，必须包含 RA、DEC、SR 三个参数；服务必须以 VOTable 标准格式返回检索结果，并且 VOTable 中必须至少含有三个独立的<FIELDS>来描述天体名称、赤经、赤纬信息^[49]。

锥形检索服务开发过程包含以下步骤：将 CXF 插件集成到 Eclipse 开发环境中；在 Eclipse 中编写服务端接口和实现接口类代码，并完成服务配置文件；将 Eclipse 中的服务打包并部署到 Tomcat 中；在浏览器或客户端中调用服务。

以下详细介绍各步骤：

（1）配置 Tomcat 和 CXF

下载最新版的 Apache CXF 包，通过设置 Web Services—CXF 2.x Preferences 将 CXF 插件集成到 Eclipse 中。下载 Tomcat 6.0，解压缩到本地。创建动态 Web 项目，并配置 Eclipse 运行环境，指定 Web service runtime 为 Apache CXF 2.x，指定 Server runtime 为 Tomcat 6.0。

（2）编写 Web 服务代码

先开发服务端 WebService 接口。再编写 WebService 实现类，其中，JAX-WS 注解@WebService()将 ConeSearch 类描述为 Web Service，以便服务发布。关键代码如图 4.4 所示。


```

@WebService(targetNamespace = "http://jx.com/", name = "ConeSearch")
public class ConeSearch {
    @WebResult(name = "return", targetNamespace = "")
    @RequestWrapper(localName = "search", targetNamespace = "http://jx.com/", className = "com.jx.Search")
    @WebMethod
    @ResponseWrapper(localName = "searchResponse", targetNamespace = "http://jx.com/", className = "com.jx.SearchResponse")
    public java.lang.String search(double ra, double dec, double sr) throws Exception
    {
        String s;
        StringBuffer str=new StringBuffer("");
        /*连接MySQL数据库*/
        String userName = "root";
        String password = "123";//数据库密码
        String url = "jdbc:mysql://localhost:3306/hxmtdata";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        Connection conn = DriverManager.getConnection (url, userName, password);
        System.out.println ("Database connection established");
        Statement stmt = conn.createStatement ();
        /*锥形检索SQL语句*/
        String DBQuery = "SELECT
EXP_ID,RA_NOM,DEC_NOM,OBJECT,OBS_MODE,INSTRUME,PROPOSAL,OBS_ID,EXPOSURE,DATE_OBS,DATE,STATUS,PATH,
HTM_NAME FROM tl_product_meta WHERE DEGREES(ACOS(SIN(RADIANS(DEC_NOM))* SIN(RADIANS("+dec+")) + COS
(RADIANS(DEC_NOM)) * COS(RADIANS("+dec+")) *COS(RADIANS(RA_NOM-"+ra+")))) <"+sr+"";
        ResultSet rs = stmt.executeQuery(DBQuery);
        ResultSetMetaData rsmd = rs.getMetaData();//元数据
        int numColumns = rsmd.getColumnCount();
        //str自加VOTable头
        str.append("<?xml version='1.0' encoding='utf-8'?'>\n");
        str.append(...);
        while (rs.next ())
        {
            str.append("    <TR>\n");
            for( int i = 1; i <= numColumns; i++ )
            {
                str.append("        <TD>"+rs.getObject(i)+"</TD>\n");
            }
            str.append("    </TR>\n");
        }
        str.append("    </TABLEDATA>\n");
        str.append("    </DATA>\n");
        str.append("  </TABLE>\n");
        str.append("</RESOURCE>\n");
        str.append("</VOTABLE>");
        conn.close ();
        System.out.println ("Database connection terminated");
        s=str.toString();
        return s;
    }
}

```

图 4.4 WebService 实现类

在此过程中，Eclipse 自动生成依赖的 bean 类，例如 ConeSearch 类对应的 bean 类为 Search 类。其中，@XmlRootElement 等 JAXB 注解，是用来根据 bean 类生成 xml 内容的。关键代码如图 4.5 所示。

```

@XmlRootElement(name = "search", namespace = "http://jx.com/")
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "search", namespace = "http://jx.com/", propOrder = {"RA", "DEC", "SR"})
public class Search {
    @XmlElement(name = "RA")
    private double RA;
    @XmlElement(name = "DEC")
    private double DEC;
    @XmlElement(name = "SR")
    private double SR;
    public double getRA() {
        return this.RA;
    }
    public void setRA(double newRA) {
        this.RA = newRA;
    }
    public double getDEC() {
        return this.DEC;
    }
    public void setDEC(double newDEC) {
        this.DEC = newDEC;
    }
    public double getSR() {
        return this.SR;
    }
    public void setSR(double newSR) {
        this.SR = newSR;
    }
}

```

图 4.5 Search 实现类

(3) 写服务配置文件

web.xml 中对 CXF 的配置如图 4.6 所示：

```

<!-- CXF的Servlet -->
<servlet>
    <description>Apache CXF Endpoint</description>
    <display-name>cxfr</display-name>
    <servlet-name>cxfr</servlet-name>
    <servlet-class>org.apache.cxf.transport.servlet.CXFServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>cxfr</servlet-name>
    <!-- 触发CXF Servlet的URL过滤器-->
    <url-pattern>/services/*</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>60</session-timeout>
</session-config>
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>WEB-INF/cxf-beans.xml</param-value>
</context-param>
<!-- Spring监听的配置 -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

```

图 4.6 web.xml 文件

cxfr-beans.xml 是 Spring 的配置文件之一。其中，implementor 表明将 ConeSearch 类发布成一个 Webservice，address 表明服务的 URL 路径，wsdlLocation 用来设置 WSDL 文件的位置。文件内容如图 4.7 所示。

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:jaxws="http://cxf.apache.org/jaxws"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-
2.5.xsd http://cxf.apache.org/jaxws http://cxf.apache.org/schemas/jaxws.xsd">
  <import resource="classpath:META-INF/cxf/cxf.xml" />
  <import resource="classpath:META-INF/cxf/cxf-extension-soap.xml" />
  <import resource="classpath:META-INF/cxf/cxf-servlet.xml" />
  <jaxws:endpoint xmlns:tns="http://jx.com/" id="conesearch"
    implementor="com.jx.ConeSearch" wsdlLocation="wsdl/conesearch.wsdl"
    endpointName="tns:ConeSearchPort" serviceName="tns:ConeSearchService"
    address="/ConeSearchPort">
    <jaxws:features>
      <bean class="org.apache.cxf.feature.LoggingFeature" />
    </jaxws:features>
  </jaxws:endpoint>
</beans>

```

图 4.7 cxf-beans.xml 文件

conesearch_schema1.xsd 文件内容如图 4.8 所示，该 XML Schema 描述了在 WebService 中使用的结构。

```

<?xml version="1.0" encoding="utf-8"?><xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://jx.com/"
  attributeFormDefault="unqualified" elementFormDefault="unqualified" targetNamespace="http://jx.com/">
  <xs:element name="search" type="tns:search"/>
  <xs:element name="searchResponse" type="tns:searchResponse"/>
  <xs:complexType name="search">
    <xs:sequence>
      <xs:element name="RA" type="xs:double"/>
      <xs:element name="DEC" type="xs:double"/>
      <xs:element name="SR" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="searchResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Exception" type="tns:Exception"/>
  <xs:complexType name="Exception">
    <xs:sequence>
      <xs:element minOccurs="0" name="message" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

图 4.8 Conesearch Schema 文件

(4) 启动 Web 应用

将该 Web 服务发布到 Tomcat 服务器之后，通过 URL 地址即可查看自定义的 WebService 接口生成的 wsdl，WSDL 文件内容如图 4.9 所示。

```

<?xml version="1.0" encoding="UTF-8">
<wsdl:definitions name="ConeSearchService" targetNamespace="http://jx.com/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://jx.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://jx.com/" schemaLocation="conesearch_schema1.xsd"/>
    </schema>
  </wsdl:types>
  <wsdl:message name="searchResponse">
    <wsdl:part name="parameters" element="tns:searchResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="search">
    <wsdl:part name="parameters" element="tns:search">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="Exception">
    <wsdl:part name="Exception" element="tns:Exception">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="ConeSearch">
    <wsdl:operation name="search">
      <wsdl:input name="search" message="tns:search">
      </wsdl:input>
      <wsdl:output name="searchResponse" message="tns:searchResponse">
      </wsdl:output>
      <wsdl:fault name="Exception" message="tns:Exception">
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ConeSearchServiceSoapBinding" type="tns:ConeSearch">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="search">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="search">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="searchResponse">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="Exception">
        <soap:fault name="Exception" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ConeSearchService">
    <wsdl:port name="ConeSearchPort" binding="tns:ConeSearchServiceSoapBinding">
      <soap:address location="http://localhost:8080/ConeSearchPort"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

图 4.9 WSDL 文件

启动 Tomcat，在浏览器中直接输入网址即可调用该锥形检索服务。例如，输入携带赤经、赤纬、检索半径三个参数的 URL 地址：
<http://192.168.111.98:8080/HXMTProject/services/ConeSearchPort/search?RA=210&DEC=54.0&SR=1.0>，服务成功接收参数并查询返回包含检索结果的标准 VOTable 文件。

3、有效性测试

采用 VAO 开发的锥形检索校验工具（Cone Search Validation，详见 <http://nvo.ncsa.uiuc.edu/dalvalidate/csvalidate.html>），可对开发的锥形检索服务是否符合 IVOA SCS 协议规范进行验证。测试结果显示为成功。由此可见开发的锥形检索服务符合 SCS 协议。

通常所说的建立 VO 兼容的 HXMT，将 HXMT 卫星数据融入 VO 环境，主要具有两方面的含义。一方面是指 HXMT 卫星数据检索发布系统能够对 VOTable 格式的数据进行解析和处理，直接访问并利用其他符合 VO 标准的天文项目数据。另一方面，HXMT 卫星数据产品能共享到 VO 环境中，成为 VO 资源的一部分。

要将 HXMT 卫星作为一个数据服务提供者，把各类数据产品共享发布到 VO 环境中，主要有两项重要工作。一是按照 VO 数据标准对数据集进行封装并按照 VO 数据访问协议提供锥形检索服务。二是按照 VO 注册标准对数据产品和相应的访问服务进行注册，以便让 VO 用户能利用 VO 资源与服务发现机制发现 HXMT 数据资源，进而访问和利用这些资源。

VO 应用环境主要包括发现、注册、发布、存储四大要素。这 4 类要素通过自由组合与相互作用构成不同的功能实体。例如，本文只选取发现、发布、存储三大要素进行功能开发与实现；美国虚拟天文台的 VAO 提供注册、发现功能；Chandra Web Chaser 数据检索发布系统实现发布、存储功能。

HXMT 数据融入 VO 应用环境的示意图如图 4.10 所示。首先，按照国际天文学界标准规范和一定的组织归档目录格式，对 HXMT 各级数据产品进行归档、存储，把全部数据实体文件存储在文件系统中，并抽取元数据信息存储到数据库中。然后，进行锥形检索服务开发，提供对外访问接口，将 HXMT 数据发布为符合 VO 标准协议的 HXMT 节点。再将 HXMT 节点注册到 VAO 注册系统

(<http://vao.stsci.edu/publishing/>) 中。注册时需提供的信息包括资源名称、简称、全球唯一标识、描述、关键字、获取方式、数据所有者信息等。VAO 注册系统收录了遵循 VO 标准协议发布的各个天文项目数据节点信息，例如 Chandra、Herschel 等天文卫星数据。HXMT 数据只有进行 VO 资源注册并传播到 VO 网络中后，才能被 VAO 等 VO 门户网站提供的资源发现服务检索浏览到。HXMT 门户网站既可直接访问底层存储的 HXMT 卫星数据资源，也可与 Aladin 等其他符合 VO 标准规范的天文项目数据资源进行互操作、互访问。因此，用户可以通过 HXMT 门户网站或 VAO 等国际门户网站检索到 HXMT 数据，同时，也可通过 HXMT 门户访问到 Aladin 中整合的天文数据资源，从而实现了 HXMT 卫星数据与国际 VO 环境中其他天文项目数据的互通互联。

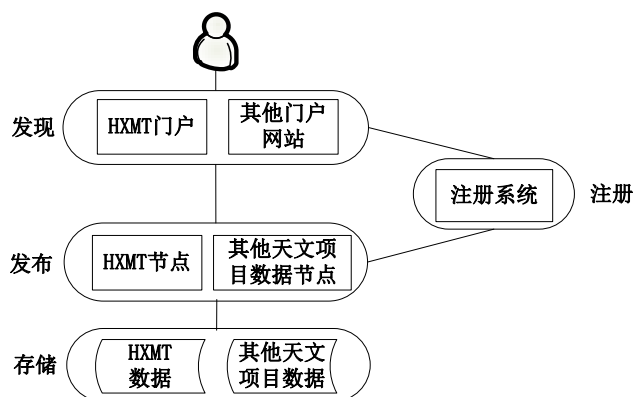


图 4.10 HXMT 卫星数据融入 VO 应用环境示意图

因为本文开发的锥形检索服务符合 SCS 协议，成功将 HXMT 数据发布为 HXMT 节点，只要再将开发的锥形检索服务注册到 VO 注册系统中，即可实现与国际 VO 环境的打通。所以，锥形检索服务的有效性间接证明了 HXMT 卫星数据融入 VO 环境的可行性。

4.2.2 数据下载

1、流程图

数据下载模块的流程图如图 4.11 所示。用户从成功检索后的检索结果显示界面勾选期望下载的数据条目，并进行后续操作。若选择“导出”功能，则由用户确定输出格式（VOTable、CSV、HTML、XSL、TXT）后，以选定的文件格式将检索结果保存至本地指定路径中；若选择“下载”功能，则访问底层相应的数据产品实体文件，并将其进行打包，最后下载到用户指定的本地目录中。

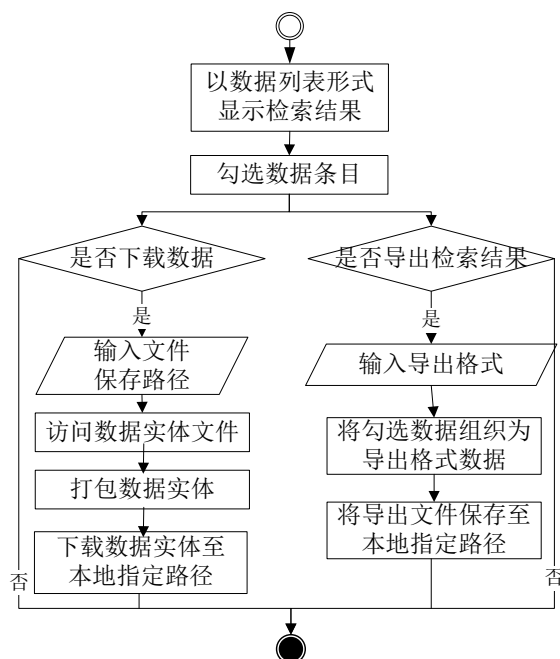


图 4.11 数据下载模块流程图

2、编码实现

由 ZipUtils 类提供打包压缩功能，ZipUtils 调用 jar 包 org.apache.tools.zip 实现 zip 压缩。关键代码如下

```
public static void doZip(ZipOutputStream zipOut, String base, String filePath) throws
Exception {
```

```
    //增加 GBK 编码设置
    zipOut.setEncoding("GBK");
    File file = new File(filePath);
    File[] files = file.listFiles();
    if (null == files) {
        return;
    }
    int len = files.length;
    FileInputStream fin = null;
    File tempFile = null;
    byte[] b = new byte[1024];
    for (int i = 0; i < len; i++) {
        tempFile = files[i];
        String tpaths = tempFile.getPath();
        String pa = tpaths.substring(base.length() + 1, tpaths.length());
        String fileName = tempFile.getName();
        String res = pa + File.separator + fileName;
        if (tempFile.isDirectory()) {
```

```

        zipOut.putNextEntry(new ZipEntry(pa + File.separator));
        doZip(zipOut, base, tempFile.getPath());
    } else {
        if (pa.equals(fileName)) {
            res = fileName;
        } else {
            res = pa;
        }
        zipOut.putNextEntry(new ZipEntry(res));
        fin = new FileInputStream(tempFile);
        int blen = 0;
        while ((blen = fin.read(b)) != -1) {
            zipOut.write(b, 0, blen);
        }
        fin.close();
    }
}
}
}

```

4.2.3 可视交互

1、流程图

可视交互模块的流程图如图 4.12 所示。(1) 若用户选择“关联检索”，则提取检索结果显示界面的目标名称信息并导入 Aladin 中。Aladin 会自动检索 Simbad 数据库，并将检索结果数据以点源显示、图片加载、星表条目自动添加的方式呈现出来。(2) 若用户选择“点源标注”，则将检索结果显示界面检索结果数据以 VOTable 格式导入到 Aladin 中。Aladin 会自动将 VOTable 中的赤经、赤纬信息以点源方式标注在 Aladin 界面上。(3) 若用户选择“可视浏览”，则由用户先勾选检索结果显示界面某行数据，系统获取选中数据对应的 FITS 格式图片文件，并导入 Aladin 中。Aladin 会展示 FITS 格式图片，以供用户浏览。

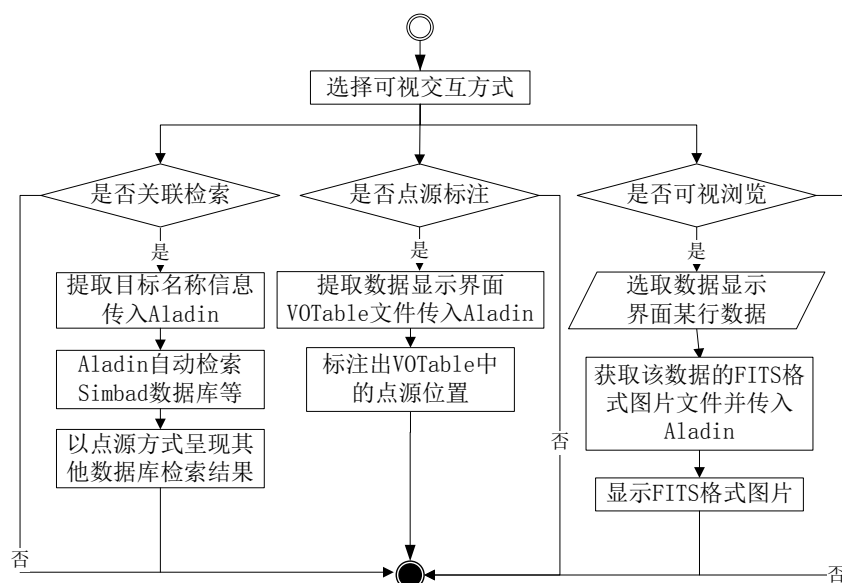


图 4.12 可视交互模块流程图

2、编码实现

Aladin 作为开源可视交互软件，提供 Java 接口、IDL 接口、CGI 程序、插件调用等方式，以实现与其他软件工具之间的数据传递与信息交互。

本系统通过调用 Aladin 提供的 Javascript 命令：

`applet.execAsyncCommand("get Aladin,Simbad "+target)`，将检索结果显示界面的目标名称信息或赤经赤纬位置坐标信息以参数形式传入 Aladin 中，Aladin 自动检索 Simbad 数据库并加载相应目标位置的图片及星表信息，从而达到可视检索效果。

通过将检索结果显示界面的检索结果信息缓存成 VOTable 文件，用一个临时的 URL 读取该 VOTable，调用 Aladin 提供的 script 命令直接读取这个 URL，即可实现在线显示检索结果的点源信息，达到点源标注效果。

同样，FITS 图片文件也可通过调用 Aladin 提供的 script 命令来读取并显示，以供用户浏览图片。

关键代码如下：

```

/*传入 target 目标名称，启动 Aladin，并检索 Simbad 数据库*/
applet.execAsyncCommand("get Aladin,Simbad "+target)
/*将 VOTable 或 FITS 图片文件以远程 URL 形式，传入 Aladin */
applet.execAsyncCommand("-script=get File(http://the.url.for.loadin/YourFile)");
/*将 VOTable 或 FITS 图片文件以本地文件形式*/
applet.execAsyncCommand("-script=get File(\"HXMT_VOTable.xml\")");

```

4.2.4 数据库实现

在 MySQL 数据库中新建名为 hxmtdata 的数据库，并建立 HXMT 卫星 1L 级数据产品实体文件信息表（1L_File_Info）、HXMT 卫星 1L 级产品元数据表（1L_Product_Meta）、HXMT 卫星 1L 级数据产品组织信息表（1L_Product_Org）、用户信息表（User_Inf）。SQL 脚本如图 4.13 所示。

```
-- MySQL dump 10.10
-- Host: localhost    Database: hxmtdata
-- Server version  5.0.22-community-nt

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table '1l_product_meta'
--
DROP TABLE IF EXISTS '1l_product_meta';
CREATE TABLE '1l_product_meta' (
  'EXP_ID' varchar(50) NOT NULL,
  'RA_NOM' text NOT NULL,
  'DEC_NOM' text NOT NULL,
  'OBJECT' varchar(10) default NULL,
  'OBS_MODE' varchar(5) default NULL,
  'INSTRUME' varchar(5) default NULL,
  'PROPOSAL' varchar(50) default NULL,
  'OBS_ID' varchar(50) default NULL,
  'EXPOSURE' text,
  'DATE_OBS' text,
  'DATE' text,
  'STATUS' varchar(10) default NULL,
  'PATH' text,
  'HTM_NAME' varchar(100) default NULL,
  PRIMARY KEY ('EXP_ID'),
  KEY 'HTMIndex' USING BTREE ('HTM_NAME')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table '1l_file_info'
--
DROP TABLE IF EXISTS '1l_file_info';
CREATE TABLE '1l_file_info' (
  'FILE_NAME' varchar(100) NOT NULL,
  'FILE_TYPE' varchar(30) NOT NULL,
  'FILE_SIZE' text NOT NULL,
  'PATH' text,
  'LEVEL' varchar(10) default NULL,
  'IMPORT_TIME' timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
  PRIMARY KEY ('FILE_NAME')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table '1l_product_org'
--
DROP TABLE IF EXISTS '1l_product_org';
CREATE TABLE '1l_product_org' (
  'EXP_ID' varchar(50) NOT NULL,
  'FILE_NAME' varchar(100) NOT NULL,
  'FILE_TYPE' varchar(30) NOT NULL,
  PRIMARY KEY ('EXP_ID','FILE_NAME'),
  KEY 'FK_1L_Product_Org_2' ('FILE_NAME'),
  CONSTRAINT 'FK_1L_Product_Org_1' FOREIGN KEY ('EXP_ID')
REFERENCES '1l_product_meta' ('EXP_ID'),
  CONSTRAINT 'FK_1L_Product_Org_2' FOREIGN KEY ('FILE_NAME')
REFERENCES '1l_file_info' ('FILE_NAME')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table 'user_info'
--
DROP TABLE IF EXISTS 'user_info';
CREATE TABLE 'user_info' (
  'USER_NAME' varchar(20) NOT NULL,
  'PASSWORD' varchar(20) NOT NULL,
  'EMAIL' varchar(20) NOT NULL,
  'ROLE' varchar(20) NOT NULL,
  'RIGHT' varchar(20) NOT NULL,
  PRIMARY KEY ('USER_NAME')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

图 4.13 数据库 SQL 脚本

4.2.5 数据库索引实现

1、流程图

采用 HTM 索引方法的锥形检索流程如图 4.14 所示。HTM 索引方法中，在对用户提供锥形检索服务之前，首先需要完成对原有数据库表结构的修改。新增表列 HTM_NAME，并对其建立普通索引。对包含（RA，DEC）空间位置坐标信息的元数据表中每一行数据进行扫描，根据每一行的（RA，DEC）值计算出相应的 HTM_NAME（即 HTM 天区单元编码），并添加新增的 HTM_NAME 列中。在建库阶段完成这些工作后，才能提供后续服务。

当用户使用锥形检索服务时，服务接收用户输入的三个参数（RA，DEC，SR），并根据这三个参数构成的圆形区域和特定的划分级次计算出该天区范围内对应的所有 HTM_NAME 值的集合 H。然后，检索数据库拣选出 HTM_NAME

值包含在上述 H 集合内的数据子集 D。因为 HTM_NAME 字段加了索引，所以查询速度比较快。最后，对刚检索出的数据子集 D 逐一使用锥形检索公式计算判断每行数据的空间位置是否锥形检索天区范围内。若存在数据行符合条件，则将最终的检索结果和元数据信息封装为 VOTable 格式文件返回给用户；若不存在，则返回固定格式的错误信息给用户。

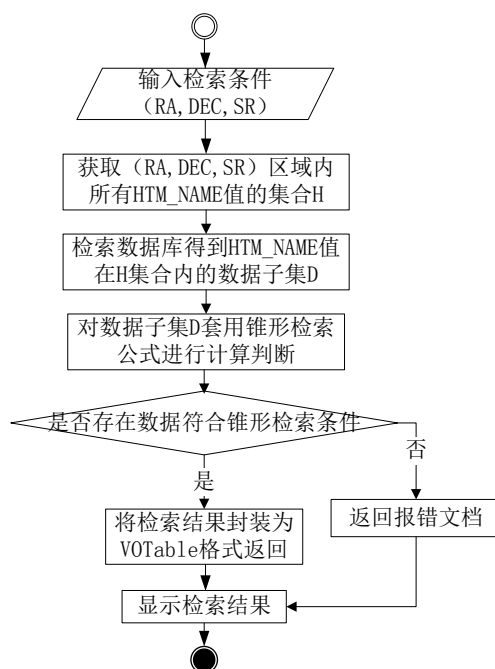


图 4.14 采用 HTM 索引方法的锥形检索流程图

2、具体实现步骤

①选择合适的划分级次。根据 HTM 天区划分方式，不同划分级次对应不同的单元数和单元大小。为了使得每个单元中星源数目不会太多，同时将单元面积控制在用户通用的 SR 范围左右，最终确定划分级次为 10。

②代码实现。调用美国约翰霍普金斯大学开发的 htmIndex.jar 包，关键代码如下。

```

/*根据赤经、赤纬，计算对应的天区单元编号HTM_NAME*/
public String getHTMName(double ra, double dec,int level) throws Exception
{
    HTMIndex si = new HTMIndexImp(level);
    String htmName = si.lookup(ra, dec);
    return htmName;
}
/*根据用户锥形检索条件(RA,DEC,SR)，计算用户检索天区内所有的 HTM_NAME 值

```

```

集合*/
    public String HTMSearch(int level,double ra, double dec,double sr) throws Exception
    {
        String result,str;
        StringBuffer sb=new StringBuffer("");
        HTMindex si = new HTMindexImp(level);
        Circle cir=new Circle(ra,dec,sr*60);
        Domain dom = cir.getDomain();//根据用户锥形检索条件(RA,DEC,SR)，生成天
区环形域
        dom.setOlevel(level);//设置划分层次
        HTMrange range = new HTMrange();
        dom.intersect((HTMindexImp)si, range,false);//得到dom区域内所有
HTM_NAME值的集合，并存放到range中
        Iterator iter = new HTMrangeIterator(range, true);
        while(iter.hasNext())
        {
            str=(String)iter.next();
            sb.append(str);
            sb.append(",");
        }
        sb.delete(sb.length()-3,sb.length()-1);
        sb.deleteCharAt(sb.length()-1);
        result=sb.toString();
        return result;
    }

```

3、性能测试

本节主要对采用 HTM 索引前后数据库系统的检索效率进行比较与分析。

(1) 环境条件

处理器：Intel(R) Core(TM)2 Quad CPU Q9300

主频：2.66GHz

内存：4.00GB

系统类型：64 位操作系统

操作系统：Windows 7

(2) 测试数据

数据库建立名为 hxmtdata 的数据库表，共存储有大约 10 万行记录。

(3) 测试结果

本实验设置 4 次不同的检索条件，分别对应 4 个实验序号，如表 4.1 所示。

表 4.1 实验检索条件

实验序号	检索条件		
1	RA: 210	DEC: 54	SR: 5
2	RA: 10	DEC: 41	SR: 5
3	RA: 83	DEC: 22	SR: 5
4	RA: 283	DEC: 10	SR: 5

4 次对比实验的运行时间均采用运行 10 次取平均值的方式获得, 测试结果记录如表 4.2 所示。

表 4.2 HTM 索引效率测试结果对比

检索时间 (ms) \ 实验序号	1	2	3	4
是否采用 HTM 索引				
是	66.7	50.8	105.3	136.1
否	198.4	173.5	325.9	427.2

4、结果分析

由测试结果可知, 应用 HTM 索引后检索性能有所提升。本次实验中查询的数据集仅为约 10 万条数据。若数据资源增至百万千万甚至上亿级, 性能提升的效果将更加显著。

若不采用 HTM 索引, 进行检索时需要对每条数据条目调用锥形检索公式查询, 数据库难以优化 SQL 语句中的三角函数计算部分, 且为避免内存溢出必须采用分段查询策略, 势必会导致检索速率底下。若采用 HTM 索引方法, 数据库中完全可以不采用分段查询, 且仅针对过滤出的子集应用锥形检索公式查询, 故提升了检索性能。

HTM 索引方法优点显著, 同时也存在着部分缺陷, 例如数据库表中必须增加含有天区单元编号的字段, 且针对含有赤经、赤纬字段的数据库表中每一行数据条目依次计算出对应的天区单元编号。这需要耗费一定的时间。但这些工作均在搭建数据库的过程中完成, 当用户使用锥形检索服务时, 无需重复这些工作, 因此采用 HTM 索引还是具有一定意义的。

4.3 系统运行

4.3.1 运行环境

系统运行时，服务器端需按下述情形进行环部署。

数据库：MySQL5.5;

Java 运行环境：JDK1.6;

Web 服务发布引擎：Tomcat 6.0。

客户端安装有 IE、Firefox 等浏览器即可，对其他硬件、存储设备等均无特殊要求。

4.3.2 运行界面

HXMT 卫星数据检索发布系统各功能模块的运行结果显示如下。

1、检索浏览

HMXT 卫星数据检索发布系统的检索浏览界面如图 4.15 所示。用户可在左面板输入检索条件进行数据检索，可在右面板检索结果显示界面进行检索结果的浏览。符合 SCS 协议的锥形检索服务返回的 VOTable 格式结果数据如图 4.16 所示。



图 4.15 HXMT 检索浏览界面

```
<?VOTable xmlns="http://www.ivoa.net/xml/VOTable/v1.2" xmlns:scs="http://www.ivoa.net/xml/STC/v1.30" xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/v1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.2">
  <RESOURCE type="results">
    <INFO value="OK" name="QUERY_STATUS"/>
    <PARAM value="ICRS" name="CoordRefFrame" arraysize="1" ucd="VOX:STC_CoordRefFrame" datatype="char"/>
    <PARAM value="TAN" name="CoordProjection" arraysize="1" ucd="VOX:WCS_CoordProjection" datatype="char"/>
    <PARAM value="X-ray" name="Bandpass" arraysize="1" ucd="VOX:BandPass_ID" datatype="char"/>
    <PARAM value="full" name="BandpassUnit" arraysize="1" ucd="VOX:BandPass_Unit" datatype="char"/>
    <TABLE name="HXMT Data">
      <DESCRIPTION>Single Cone Search for HXMT Data Explanations and Statistics of UCDs: http://www.ivoa.net/documents/REC/UCD/UCDlist-20070402.html</DESCRIPTION>
      <FIELD name="obsid" ucd="meta.id" datatype="int">
        <DESCRIPTION>HXMT observation identifier</DESCRIPTION>
        <VALUES null="2147483648"/>
      </FIELD>
      <FIELD name="ra" ucd="pos.eq.ra" datatype="double" unit="deg">
        <DESCRIPTION>Right ascension, ICRS(J2000)</DESCRIPTION>
      </FIELD>
      <FIELD name="dec" ucd="pos.eq.dec" datatype="double" unit="deg">
        <DESCRIPTION>Declination, ICRS(J2000)</DESCRIPTION>
      </FIELD>
      <FIELD name="target_name" arraysize="1" ucd="meta.idsrc" datatype="char">
        <DESCRIPTION>Target name</DESCRIPTION>
      </FIELD>
      <FIELD name="instrument" arraysize="1" ucd="instr" datatype="char">
        <DESCRIPTION>Instrument name</DESCRIPTION>
      </FIELD>
      <FIELD name="proposal_num" arraysize="1" ucd="obs.proposal" datatype="char">
        <DESCRIPTION>Proposal number, including several related observations</DESCRIPTION>
      </FIELD>
      <FIELD name="observation_mode" arraysize="1" ucd="obs.param" datatype="char">
        <DESCRIPTION>Observation mode, including Pointing Survey and Area</DESCRIPTION>
      </FIELD>
      <FIELD name="observation_start_time" arraysize="1" ucd="time.startobs" datatype="char">
        <DESCRIPTION>Start time of observation</DESCRIPTION>
      </FIELD>
      <FIELD name="exposure_time" ucd="time.duration:obs.exposure" datatype="double" unit="s">
        <DESCRIPTION>Total exposure time</DESCRIPTION>
      </FIELD>
      <FIELD name="publishing_status" arraysize="1" ucd="meta.code.status" datatype="char">
        <DESCRIPTION>Publishing status: unpublished, published</DESCRIPTION>
      </FIELD>
      <FIELD name="publishing_date" arraysize="1" ucd="time.release" datatype="char" type="adq:TIMESTAMP">
        <DESCRIPTION>Date/time when data is available to the public</DESCRIPTION>
      </FIELD>
    </TABLE>
    <TABLEDATA>
      <TR>
        <TD>5300</TD>
        <TD>210.756744</TD>
        <TD>54.385403</TD>
        <TD>POINTING</TD>
        <TD>ACIS</TD>
        <TD>05600587</TD>
        <TD>Dr. K.D. Kuntz</TD>
        <TD>52093.500658</TD>
        <TD>2004-03-07T09:29:06</TD>
      </TR>
      <TR>
        <TD>5340</TD>
        <TD>210.756744</TD>
        <TD>54.385403</TD>
        <TD>POINTING</TD>
        <TD>ACIS</TD>
        <TD>05600587</TD>
        <TD>Dr. K.D. Kuntz</TD>
        <TD>52093.500658</TD>
        <TD>2004-03-07T09:29:06</TD>
      </TR>
      <TR>
        <TD>6170</TD>
        <TD>210.756744</TD>
        <TD>54.385403</TD>
        <TD>POINTING</TD>
        <TD>ACIS</TD>
        <TD>05600587</TD>
        <TD>Dr. K.D. Kuntz</TD>
        <TD>52093.500658</TD>
        <TD>2004-03-07T09:29:06</TD>
      </TR>
    </TABLEDATA>
  </RESOURCE>
</VOTable>
```

图 4.16 符合 SCS 协议的 VOTable 格式锥形检索结果数据

2、数据下载

数据下载运行界面如图 4.17 所示。用户可在检索结果显示界面点击“下载实体文件”按钮，完成对勾选数据的批量下载。可点击每个数据条目右侧的下载按钮，进行单条数据下载。可点击“导出 VOTable”按钮将检索结果数据导出并保存为 VOTable 格式文件。

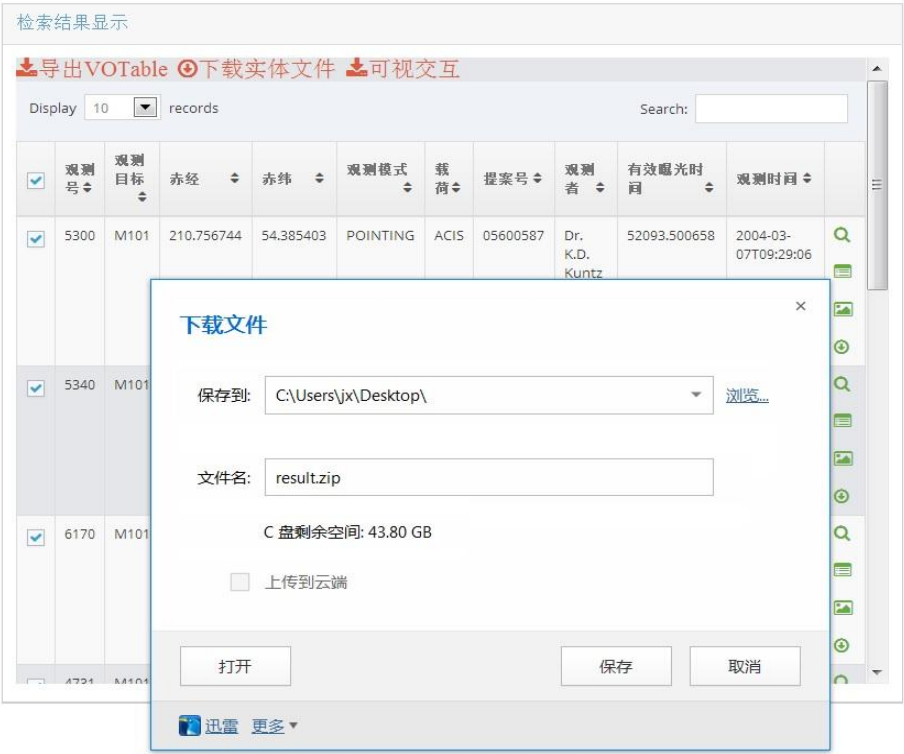


图 4.17 HXMT 数据下载界面

3、可视交互

点击检索结果显示界面的“可视交互”按钮，选择可视交互方式，即可对用户勾选数据进行关联检索、点源标注和可视浏览。也可通过点击每个数据条目右侧的三个小图标，依次启动关联检索、点源标注和可视浏览功能。关联检索运行界面如图 4.18 所示。点源标注运行界面如图 4.19 所示。可视浏览界面如图 4.20 所示。

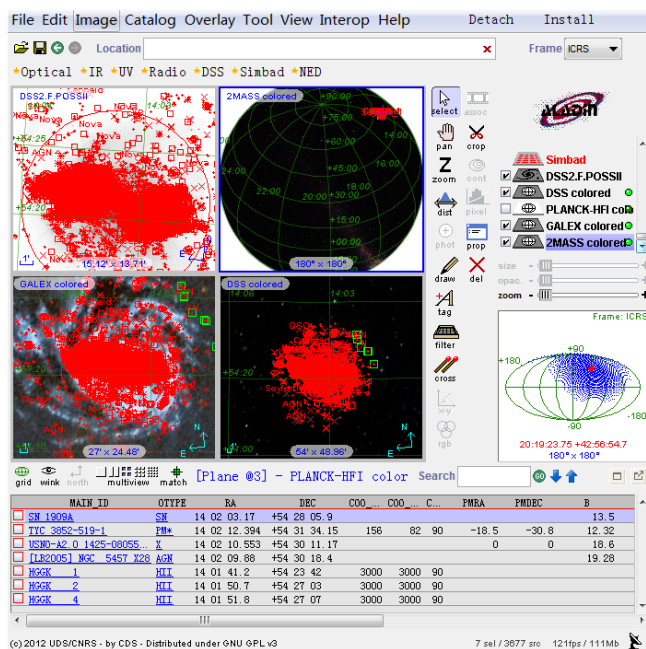


图 4.18 Aladin 可视检索界面

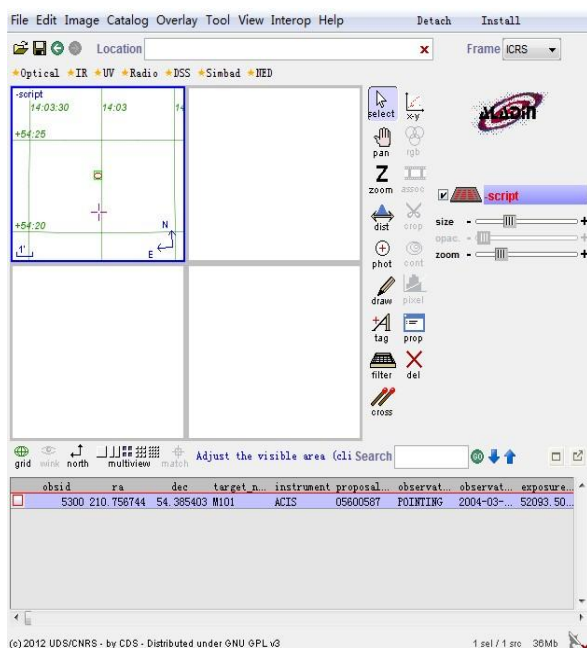


图 4.19 Aladin 点源标注界面

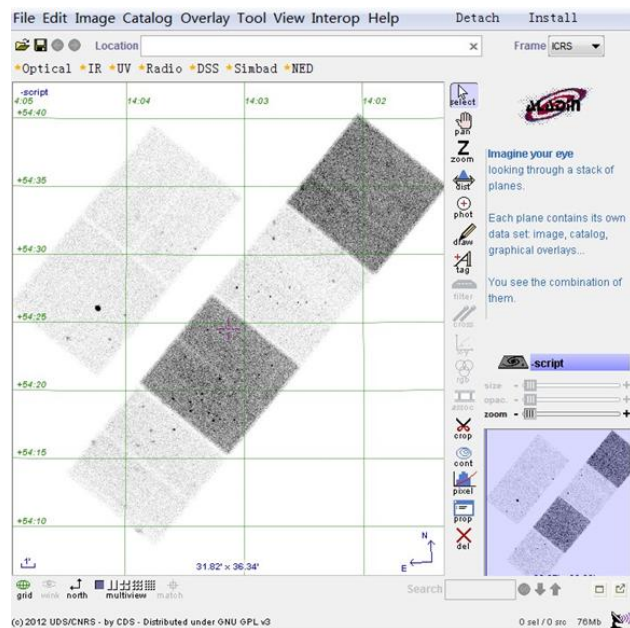


图 4.20 Aladin 可视浏览界面

4.4 本章小结

本章基于前一章的系统设计工作，重点阐述了 VO 框架下 HXMT 卫星数据检索发布系统的原型研制与实现过程。首先，确定系统开发环境，包括开发语言、Web 服务容器、数据库等，并确定采用 Spring 框架进行系统研制，采用 SOAP 样式 Web Service 技术开发锥形检索服务。其次，详细阐明了检索浏览、数据下载、可视交互三大功能模块的实现方法，并成功验证了开发的锥形检索服务的有效性。然后，具体描述了建立 HTM 数据库索引的步骤，并对引入 HTM 索引算法前后的系统检索效率进行性能测试对比。最后依次展示了各功能模块的运行效果。

第五章 总结与展望

5.1 工作总结

HXMT 卫星数据检索发布系统为中国首颗空间天文卫星——HXMT 卫星建立起了一个集检索浏览、数据下载和可视交互功能于一体的用户友好型系统。本文完成的工作主要包括以下四方面。

1、研究了虚拟天文台基础理论与技术框架

深入调研了国际上虚拟天文台的产生背景、概念定义、制定的标准协议以及国内外现阶段的发展情况,分析得出在系统研制过程中引入 VO 技术具有一定的必要性。从体系结构、数据格式和数据访问协议三个方面,归纳梳理了常用的 VO 标准协议与规范,包括 IVOA 概念体系结构,VOTable 数据互访问格式,SCS、SIA、TAP 等数据访问协议。为融入 VO 环境,HXMT 卫星数据检索发布系统采用了 VO 标准与协议中的 IVOA 概念体系结构、VOTable 数据格式、UCD 语义标准和 SCS 数据访问协议。

2、进行了 HXMT 卫星数据组织管理

参与了 HXMT 卫星数据组织管理工作。按数据处理程度将 HXMT 卫星数据产品分为 0 级、1 级(1L、1S)和 2 级(2L)。针对公开发布的 1L、1S 和 2L 级数据产品设计了合理的归档目录结构。参照国际空间天文领域数据标准规范,采用了 FITS 格式存储数据实体文件。最后,确定了数据存储格式,并梳理了数据产品种类。从而实现了 HXMT 卫星数据合理高效的归档管理。

3、设计了 HXMT 卫星数据检索发布原型系统

基于对虚拟天文台标准协议及应用服务的研究分析,完成了 HXMT 卫星数据检索发布系统的原型设计。首先结合引入 VO 技术发布 HXMT 卫星数据的功能需求,兼顾考虑了提供用户友好型界面、提升检索速率的性能需求,从而明确了检索浏览、数据下载、可视交互三大核心功能模块;然后参考 IVOA 体系结构,构建一套由资源层、服务层、用户层三层组成的体系架构;最后,针对 HXMT 卫星数据量大、物理特性复杂、衍生产品多等特点,设计了数据实体文件、元数

据信息、互访数据分离的分层数据模型来管理和使用数据，并详细阐述了数据库的概念设计、逻辑设计与物理设计，论证了采用 HTM 索引方法的可行性。

4、实现了 HXMT 卫星数据检索发布原型系统

采用 Spring 框架和 WebService 技术，研发并实现了基于 VO 技术框架的 HXMT 卫星数据检索发布原型系统。该原型系统拥有以下四方面的特点：

(1) 遵循 SCS 协议、VOTable 数据格式、UCD 语义标准，开发了锥形检索服务。该锥形检索服务成功通过了 VO 有效性验证，经 VO 注册后即可将 HXMT 数据资源发布到 VO 环境中，从而间接证明了 HXMT 数据融入 VO 环境的可行性。有助于 HXMT 卫星与国际各天文项目之间的数据共享，有力地推动了 VO 技术在实际天文项目中的应用。

(2) 原型系统提供了直接与用户交互的 HXMT 门户网站，集成了检索浏览、数据下载和可视交互等应用功能，成功完成了调用 Aladin 开源软件进行关联检索、点源标注和可视浏览的功能。

(3) 原型系统采用了 HTM 索引算法优化数据库结构，将二维的空间参量映射到一维参数空间，降低检索维度，从而大大加快检索速率。通过测试实验进行性能比对，得出 HTM 索引能有效提升天文数据检索效率的结论。这对处理海量天文数据检索、提升检索性能具有重要的意义。

(4) 原型系统采用松耦合的 MVC 模式和 Spring 框架进行软件开发，基于 SOA 架构和 Web Service 技术进行 SOAP 样式 Web 服务开发，降低组件间的依赖程度，增加了代码的可读性，具有良好的灵活性和扩展性。

5.2 创新点

本文的创新点主要体现在以下几个方面：

1、HXMT 卫星数据组织管理

针对 HXMT 卫星数据物理特性复杂、数据量大、衍生产品多等特点，详细充分研究了 HXMT 卫星数据，包括进行数据产品分级，设计归档目录结构，确定数据存储格式以及数据产品种类等。不仅实现了对 HXMT 卫星数据合理高效的归档管理，而且对今后我国其他空间天文卫星数据的归档管理具有良好的参考

价值。

2、引入虚拟天文台技术

将虚拟天文台技术引入到我国首颗空间天文卫星 HXMT 数据检索发布系统的原型搭建与研发过程中。对 IVOA 概念体系架构进行了适当裁剪,遵循 SCS 数据访问协议、VOTable 数据格式和 UCD 语义标准,研发了符合 VO 规范的 HXMT 卫星数据检索发布系统。该系统既为科研人员及天文爱好者提供了专门检索浏览 HXMT 卫星各级科学数据产品的应用平台,又间接证明了将 HXMT 数据发布到 VO 环境的可行性。有利于 HXMT 项目与国际上不同天文项目之间数据的互访问与互操作,同时也为虚拟天文台的发展做出了一定的贡献。

3、成功设计并实现了 HXMT 卫星数据检索发布原型系统

在技术实现方面,采用 CXF 框架、遵循锥形检索(SCS)协议开发了 SOAP 样式的锥形检索服务。开发的 Web 检索服务既符合 VO 协议规范,又符合业界通用的 Web 服务标准,具备一定的扩展性和适用性。HXMT 卫星数据检索发布原型系统的成功搭建也为后续研制其他空间天文卫星数据检索发布系统提供技术支撑。

5.3 工作展望

本文已经完成基于 VO 的 HXMT 卫星数据检索发布系统原型设计与实现,对我国其他空间天文卫星数据检索发布系统的研制具有良好的参考意义。但在如下几个方面仍有待于进一步探索与研究。

1、本文虽引入了 HTM 索引来提升系统检索性能,但随着用户访问量增加以及数据产品数量的海量增长时,数据检索效率和资源下载速度势必会降低。未来可以考虑采用负载均衡技术来提升服务器响应效率及系统运行性能,同时也可引入云计算和云存储技术^[50],通过调用云服务来实现跨时空的大规模数据处理,提升海量数据检索与存储能力等,突破目前天文数据检索浏览和处理存储过程中出现的一系列瓶颈问题。

2、本文仅实现了初步的原型系统,没有对实际网络条件下的产品检索和传输性能给出评估以及技术优化方案,在后续工作中将进行补充完善。

3、本文在可视交互方面形式比较单一，后续工作会集成更多可视交互组件，以丰富可视交互效果，提升用户交互体验。例如，将 GoogleSky 嵌入到前端数据检索界面，提供星空图交互检索方式，即用户在 GoogleSky 星空图中圈画检索范围，系统可自动提取检索位置参数并自动完成锥形检索，从而建立良好的用户交互体验。

4、文本仅实现了 SCS 协议，未来针对图片、光谱等不同的数据产品种类，可遵循更多的 VO 协议（如 SIA、SAA、TAP），开发相应的图片检索服务、光谱检索服务、表检索服务等 Web 服务，以提供数据检索访问功能。

5、可添加交叉证认、ADQL 检索等功能，这将逐渐模糊数据发现和分析之间的界限，完成不同数据归档系统的数据融合与数据挖掘^[51]工作，有利于提高充分挖掘天文数据的科学价值。

参考文献

- [1]李惕碚,吴枚.空间硬 X 射线调制望远镜[J].物理,2008,37(9):648-651
- [2]姜旭, 邹自明, 佟继周. 虚拟天文台技术在 HXMT 卫星任务中应用的设想[J]. 科研信息化技术与应用, 2013, 4(006): 41-47.
- [3]李长华, 崔辰州, 赵永恒. 虚拟天文台的理念, 设计与应用[J]. 中国教育网络, 2012 (6): 22-24.
- [4]傅衍杰.空间科学虚拟观测台体系结构研究[D].北京:中国科学院研究生院, 2011.40-41.
- [5]汪惺惺. LAMOST 科学计算云平台系统的构建与应用[D]. 山东大学, 2013.
- [6]Strauss M A, Tyson J A, Anderson S F, et al.2009,preprint(astro-ph/0903.3149.
- [7]桑健. 中国虚拟天文台的数据库和数据服务[D]. 中国科学院研究生院, 2004.
- [8]李建, 崔辰州, 何勃亮, 等. 天文数据库回顾与展望[J]. 天文学进展, 2013, 1: 002.
- [9]高丹, 张彦霞, 赵永恒. 中国虚拟天文台交叉认证工具的开发和应用[J]. 天文学报, 2009 (3): 348-358.
- [10]李文. 虚拟天文台环境下的海量数据存储与访问技术研究[D]. 天津大学, 2007.
- [11]赵永恒. 互联网时代的天文学革命:虚拟天文台[J]. 科学(上海), 2002, 54(2):13-17.
- [12]赵永恒, 崔辰州. 从虚拟天文台到天文信息学[J]. 科研信息化技术与应用, 2011, 2: 3-12.
- [13]李建, 崔辰州, 何勃亮. 基于虚拟天文台的天文数据库[J]. 科研信息化技术与应用, 2011, 2: 75-80.
- [14] IVOA. Introduction to VO Concepts[EB/OL].http://www.ivoa.net/deployers/intro_to_vo_concepts.html.
- [15]赵永恒, 崔辰州. 从虚拟天文台到天文信息学[J]. 科研信息化技术与应用, 2011, 2(3): 3-12.
- [16]崔辰州, 赵永恒. 中国虚拟天文台研发策略与重点[J]. 天文研究与技术: 国家天文台台刊, 2004, 1(3): 203-209.

- [17]傅衍杰, 邹自明, 佟继周.空间科学虚拟观测台体系结构研究[J].天文研究与技术, 2011, 8(4): 395-402.
- [18]汪晶晶, 季凯帆, 王锋, 等. AstroGrid CEA 架构的应用研究[J]. 天文研究与技术: 国家天文台台刊, 2012, 9(003): 323-328.
- [19]崔辰州, 赵永恒.中国虚拟天文台体系结构[J].天文研究与技术, 2004, 1(2):140-151.
- [20]李绯, 王维国, 罗泽, 等. 集成网格服务的中国虚拟天文台 workflow 系统研究与设计[J]. 微电子学与计算机, 2008 (z1): 101-104.
- [21]崔辰州,李文,于策,徐祯,赵永恒,于建军.FITS 数据文件的检索和访问[J]. 天文研究与技术, 2008, 5(2): 116-123.
- [22] Plantea R L, Greeneb G, Hanischb R J, etal. Building archives in the virtual observatory era[J]. SPIE, 2010(7740).
- [23]张月梅.锥形检索 Web 服务的 REST 式设计和实现[D].天津大学, 2009.
- [24]李建华, 陈松乔, 马华. 面向服务架构参考模型及应用研究[J]. 计算机工程, 2006, 32(20): 100-102.
- [25]孙珊珊. 基于 SOA 的电子政务系统研究与应用[J]. 2009.
- [26]梁爽. 基于 SOA 的云计算框架模型的研究与实现[J]. 计算机工程与应用, 2011, 47(35): 92-94
- [27]项高友. 基于语义的 Web 服务发现框架实现研究[D]. 南京航空航天大学, 2009.
- [28]潘海兰, 吴翠红, 葛晓敏. XML 及其在 MVC 模式中的应用[J]. 计算机技术与发展, 2010 (2): 202-205.
- [29]任中方, 张华, 闫明松, 等. MVC 模式研究的综述[J]. 计算机应用研究, 2004, 10(528): 12.
- [30]郑程. 空间天气模式集成框架设计与原型实现[D]. 北京:中国科学院空间科学与应用研究中心, 2012. 8-10
- [31]Rod Johnson, Juergen Hoeller. Expert One-on-One JavaEE Development without EJB[J]. Wrox, June, 2004, 3(2): 103-112.
- [32]田恺. Spring 事务管理[J]. 大众商务 , 2009(6).

- [33]管江淮. 基于 Java EE 的购物系统的设计与实现[D]. 电子科技大学, 2009.
- [34]楼明珠. 基于 SSH 框架技术的网上花店系统的设计与实现[D]. 南昌大学, 2011.
- [35]孙长明, 余立建. 基于 Hibernate+ Spring+ Struts 的 Web 应用开发框架[J]. 铁路计算机应用, 2007, 16(3): 31-33.
- [36]胡启敏, 薛锦云, 钟林辉. 基于 Spring 框架的轻量级 J2EE 架构与应用[J]. 计算机工程与应用, 2008, 44(5).
- [37] (英) Matthew J G 等, 崔辰州等译. 虚拟天文台: 天文学研究的工具与技术 [M]. 北京: 中国科学技术出版社, 2010: 24-27, 81-86.
- [38]傅衍杰. 空间科学虚拟观测台体系结构研究[D]. 北京: 中国科学院研究生院, 2011. 40-41.
- [39]姚旭. 有关数据库查询优化技术在局域网中的应用研究[J]. 企业文化, 2012 (2).
- [40]赵青. 面向海量数据的高效天文交叉证认的研究[D]. 天津大学, 2010.
- [41] Kalpakis K, Riggs M, Pasad M, et al. A System for Low-Cost Access to Very Large Catalogs[C]//Astronomical Data Analysis Software and Systems X. 2001, 238: 133.
- [42]Clive Page. Indexing the Sky[EB/OL].
<http://wiki.astrogrid.org/bin/view/Astrogrid/SkyIndexing>.
- [43]Sky Server Support Team. Hierarchical Triangular Mesh[EB/OL].
http://www.skyserver.org/HTM/Old_default.aspx.
- [44]崔丽, 周相兵. 一种语义 Web 服务开发框架整合与应用[J]. 科学技术与工程, 2012, 25.
- [45]张宇, 王映辉, 张翔南. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程, 2010, 36(4): 59-62.
- [46]李晓科. 基于天文数据的服务计算与服务封装技术研究[D]. 云南省昆明市: 昆明理工大学, 2012. 21-23.
- [47]李帅. 智能代理中面向动态服务聚合的模型、算法的研究与实现[D]. 北京市: 北京邮电大学, 2010.
- [48]Andre Schaaff, Matthew Graham. IVOA Web Services Basic Profile [EB/OL].

<http://www.ivoa.net/Documents/WSBasicProfile/> .

[49]李晓科, 季凯帆, 邓辉, 等.面向 Web 服务的天文数据发布技术[J].天文研究与技术, 2012, 9(1):70-77.

[50]汪晶晶. 基于 MapReduce 的天文数据处理方法与应用研究[D]. 昆明理工大学, 2012.

[51]张彦霞, 赵永恒. 数据挖掘技术在天文学中的应用[J]. 科研信息化技术与应用, 2011, 2: 13-27.

发表文章目录

期刊论文:

- [1]姜旭, 佟继周, 崔辰州, 邹自明.基于虚拟天文台的 HXMT 卫星数据检索发布系统设计与实现[J].天文研究与技术, 2014. (已接收, 待发表)
- [2]姜旭, 邹自明, 佟继周.虚拟天文台技术在 HXMT 卫星任务中应用的设想[J].科研信息化技术与应用, 2013, 4(6):41-47.

致谢

时光荏苒，不知不觉中三年的研究生生涯即将结束。在这里留下了许多的欢笑与汗水，也使我与众多的老师和同学结下了深厚的情谊。这一切让我体悟到了成长的足迹，感觉一切刚刚开始，却又要匆匆结束。值此毕业论文完成之际，我要对三年中无私地教导关心我的老师，以及曾经一起为梦想奋斗、相互帮助支持的朋友致以衷心的感谢。

首先，要感谢我的导师佟继周老师。佟老师严谨的科研态度，敏锐的学术洞察力，一丝不苟的工作作风，都使我印象深刻、受益良多。从选题调研到论文撰写的每一个环节，都得到了佟老师的悉心指导与帮助。在科研学习的过程中，佟老师也给予我很多宝贵的建议，让我领悟到很多思考问题和解决问题的方法，使我受益匪浅。佟老师对我的关怀与帮助不仅停留在学习研究上，更体现在生活的方方面面。佟老师温柔包容的性格以及乐观豁达的人生态度，深深地激励着我不断前行。在此谨向佟老师致以诚挚的感谢和崇高的敬意。

其次，还要感谢邹自明老师对我工作学习上无微不至的指导与帮助，邹老师引导我自学所需知识，并加以灵活应用，使我明白了从实践中掌握新知的益处，在此衷心感谢邹老师的认真负责与耐心指导。

感谢实验室的师兄郑岩、郑慧东、闫振中、马福利、张垚钢、熊森林，学姐马文臻、姜文凭、陈玲、袁雅琴、胡晓彦、高文健，同学钟佳，师弟高广大、刘春蔚、康栋贺，与你们一起学习和研究让我获益良多。

感谢我的室友刘雯、朋友刘雅琪，因为有你们朝夕相处的陪伴，三年的生活才如此的舒心美好。

感谢我的男友董一伯，感谢你给予的照顾与呵护，理解与支持。此生，你若安好，便是晴天。

感谢含辛茹苦抚养我成人的父母，和从小陪伴着一起成长的弟弟，你们的关怀和鼓励是我坚持不懈的动力。愿弟弟金榜题名，愿父母身体健康。

感谢所有在百忙之中参加论文评审与答辩的老师们。

本文应用的数据来源于中国科学院信息化专项科技数据资源整合与共享工程重点数据库项目“空间科学学科数据库”及国家科技基础条件平台地球系统科学数据共享平台——“空间科学数据共享平台”，感谢其提供的数据支持。

最后，谨以此文献给所有理解、关心、支持和帮助过我的老师、同学和朋友。
感谢你们陪我走过的青春岁月！祝愿每个人工作顺利、一切平安！