

分类号_____

密级_____

UDC _____

编号_____

中国科学院研究生院

硕士学位论文

虚拟天文台数据访问服务注册系统的设计与实现

路勇

指导教师 罗阿理 副研究员 中国科学院国家天文台

赵永恒 研究员 中国科学院国家天文台

申请学位级别 硕士 学科专业名称 天文技术与方法

论文提交日期 2007 年 5 月 论文答辩日期 2007 年 6 月

培养单位 中国科学院国家天文台

学位授予单位 中国科学院研究生院

答辩委员会主席 邓李才

Design and Implementation of VO-DAS Registry System

Yong Lu

Advisor:

Dr. Ali Luo & Prof. Yongheng Zhao

National Astronomical Observatories, Chinese Academy of Science

June 2007

摘 要

虚拟天文台是天文学飞速发展并且和信息科学结合的产物，它的宗旨是将世界范围内主要天文研究资源无缝透明地整合在一起。我国从 2002 年提出并开始研究开发中国虚拟天文台项目，它的主要科学目标是给天文学家提供平台让他们方便的获取、处理并分析各国的数据，尤其是大样本数据，并最终把数据的结果展现给他们。中国虚拟天文台项目组于 2005 年开始进行虚拟天文台数据访问服务（VO-DAS）的研究。VO-DAS 能够实现异地异构数据库的访问，数据资源的自动发现，海量数据的访问与查询，多种数据格式的访问，天文数据的交叉认证等，是一个功能强大的数据访问平台。

虚拟天文台数据访问服务注册系统（VO-DAS Registry）是 VO-DAS 的重要组成部分，它为 VO-DAS 获取数据资源的元数据，从而使得 VO-DAS 更好的管理和应用这些数据资源。在获取资源元数据的过程中，此系统符合 IVOA 的元数据标准和注册接口标准，具备互操作性，能够与一般的虚拟天文台注册系统（VO Registry）进行交互。此系统目前已经成功地与 AstroGrid Registry 以及 STScI/JHU NVO Registry 进行了连接。本文先对此系统涉及到的背景概念以及此系统的基础----IVOA 的相关协议进行了研究，然后着重对 VO-DAS Registry 系统的设计和实现等作了介绍。

另外，在虚拟天文台的数据工作方面，本人还针对光谱和图像的联合查询、海量星表数据处理以及 FITS 文件入库系统进行了研究。所做的工作也一并写在本文中。其中，星表和图像的联合查询基于 IVOA 数据访问层的简单图像访问协议和简单光谱访问协议，涉及了虚拟表的概念，是一种对天文数据查询语言（ADQL）的应用和扩充的思想；海量星表数据处理和 FITS 文件入库两个工具都是从星表数据的有效使用着眼，海量星表数据处理基于数据库，能够将 ASCII 格式的星表自动化地进行数据入库、交叉认证以及数据提取工作，FITS 文件入

库工具则能将二进制格式的星表提取用户感兴趣的参数，并转换为 ASCII 格式，调用海量星表数据处理工具将星表导入数据库。

关键词： 数据注册 虚拟天文台 天文数据库 星表数据处理

Abstract

With the technological advances in astronomy and the breakthroughs in computer and Internet technology, Virtual Observatory (VO) is brought forward. The principal goal of VO is to integrate and utilize the worldwide major astronomical research resources transparently. In 2002, China-VO was launched. A virtual observatory (VO) is a collection of interoperating data archives and computing & analyzing tools, which utilize the Internet to form a scientific research environment in which astronomical research programs can be conducted. In 2005, China-VO began the research of the virtual observatory data access service (VO-DAS). VO-DAS is a grid solution which provides a way to seamlessly access large distributed and multi-waveband data resources for astronomers. VO-DAS Registry is an important sub-system of VO-DAS.

The VO-DAS Registry System serves VO-DAS as it gets and manages the metadata of both DataNode and tables of DataNode, which makes it possible to discover DataNodes automatically by VO-DAS. Furthermore, with these metadata, VO-DAS can parse queries from users to sessions it will run. As a matter of fact, VO-DAS Registry helps VO-DAS in the respect it improves the query efficiency. We designed the VO-DAS Registry based on related IVOA standards, so it has good interoperability. At present, the VO-DAS Registry System has connected with AstroGrid Registry and STScI/JHU NVO Registry. In this thesis, the technical background and related IVOA standards are described, then the design and implement of this system are introduced in detail.

Besides, in the field of VO data processing, I did some research including the unite-query of spectrums and images, the data processing of huge catalogue data and the FITS-ASCII tool. These works are also described in this thesis.

Key words: VO-DAS Registry, Virtual Observatory, Catalogue Data Processing,
Techniques: miscellaneous; Methods: miscellaneous; Astronomical data bases:
miscellaneous

目 录

1 引言	1-10
1.1 虚拟天文台研究背景	1-10
1.1.1 虚拟天文台概述	1-10
1.1.2 国内外研究现状	1-11
1.2 中国虚拟天文台概述	1-14
1.2.1 CHINA-VO 的研究领域	1-14
1.2.2 中国虚拟天文台的科学目的与意义	1-15
1.3 虚拟天文台数据访问服务 (VO-DAS) 的研究意义	1-15
2 虚拟天文台数据标准	2-1
2.1 天文数据的特点	2-1
2.2 可扩展标记语言 (XML)	2-2
2.2.1 简介	2-2
2.2.2 XML 文档	2-3
2.2.3 架构 (XML SCHEMA)	2-5
2.2.4 范例	2-6
2.2.5 XML 数据处理	2-7
2.3 VOTABLE 天文表列数据的封装	2-8
2.4 ADQL 数据查询标准	2-9
3 虚拟天文台注册系统 (VO REGISTRY)	3-1
3.1 国际虚拟天文台联盟制订的相关标准	3-1
3.1.1 资源元数据标准	3-1

3.1.2	REGISTRY 访问接口协议	3-8
3.2	各个 VO REGISTRY 介绍	3-11
3.2.1	ASTROGRID REGISTRY	3-13
3.2.2	NVO REGISTRY	3-16
4	虚拟天文台数据访问服务 (VO-DAS) 注册系统	4-1
4.1	VO-DAS 系统以及 VO-DAS REGISTRY 在其中的地位和功能	4-1
4.2	系统设计	4-3
4.3	与 VO REGISTRY 的连接	4-8
4.3.1	与 ASTROGRID REGISTRY 的连接	4-8
4.3.2	与 STSCL/JHU NVO REGISTRY 的连接	4-9
5	其他相关工作	5-1
5.1	图像和光谱的联合查询	5-1
5.1.1	简单图像访问协议 (SIA) 与简单光谱访问协议 (SSA)	5-1
5.1.2	设计	5-3
5.2	海量星表数据处理工具的自动化	5-4
5.2.1	海量天文数据处理工具的背景和简介	5-4
5.2.2	整合和将海量星表数据处理工具自动化的意义	5-5
5.2.3	设计	5-7
5.3	FITS 文件入库系统的设计	5-11
5.3.1	意义	5-11
5.3.2	设计与实现	5-12
5.3.3	流程	5-13
6	总结和展望	6-1

图 表 目 录

图表 1 各国虚拟天文台项目	1-12
图表 2 一个典型的XML架构例子	2-7
图表 3 斯隆数字巡天的资源元数据	3-8
图表 4 各种Registry的交互	3-10
图表 5 几个VO Registry的基本信息	3-11
图表 6 欧洲各国VO的Registry系统	3-12
图表 7 亚洲、美洲各虚拟天文台的Registry系统	3-13
图表 8 AstroGrid Registry系统模型	3-13
图表 9 AstroGrid Registry资源浏览页面	3-14
图表 10 AstroGrid Registry 资源搜索页面	3-15
图表 11 在AstroGrid Registry注册了的DAS资源例子	3-16
图表 12 STScI/JHU NVO Registry 资源浏览页面	3-17
图表 13 STScI/JHU NVO Registry资源注册页面	3-18
图表 14 VO-DAS Registry在VO-DAS中的功能	4-2
图表 15 VO-DAS Registry系统设计(1)	4-5
图表 16 VO-DAS Registry系统设计(2)	4-6
图表 17 VO-DAS Registry 流程示意图	4-7
图表 18 VO-DAS Registry 与其它模块交互图	4-8
图表 19 与AstroGrid Registry连接程序的一段代码	4-9
图表 20 与STScI/JHU NVO Registry连接的一段代码	4-10
图表 21 一个使用SIA 服务的查询结果 (1)	5-2
图表 22 一个使用SIA 服务的查询结果 (2)	5-3
图表 23 海浪星表数据处理工具的架构	5-6
图表 24 海量星表数据处理工具的流程图	5-11
图表 25 FITS文件入库工具流程图	5-13

1 引言

1.1 虚拟天文台研究背景

1.1.1 虚拟天文台概述

随着探测器和空间技术的发展，天文学已进入全波段巡天阶段，天文观测从可见光、射电波段扩展到包括红外、紫外、X 射线和 γ 射线在内的电磁波各个波段，多波段数据急剧增长。举例而言：目前哈勃空间望远镜（Hubble Space Telescope, HST）每天大约产生 5GB 的数据；我国正在建造的大天区面积多目标光纤光谱望远镜（The Large Sky Area Multi-Object Fiber Spectroscopic Telescope Project, LAMOST）也将每天产生 3GB 的数据；而美国计划建造的“大口径综合巡天望远镜（The Large Synoptic Survey Telescope, LSST）”，每天的观测数据将达到 18TB 的量级。

如果能有方便地获取、组织和分析这些多波段、异地、异构的天文数据的方法，利用如此庞大的数据和先进的计算机技术，就可以获得全新的天文台的研究方法进而产生大量、无法预计的新发现。

而与天文学观测和数据获取技术飞速发展的同时，正在经历迅速变革的信息技术领域正在越来越深入地改变着社会的各个领域。特别值得注意的是互联网技术和网格技术的发展，“模拟实现高性能计算机”的技术，这些技术的目标是实现互联网上所有资源的互联互通，包括计算资源、存储资源、通信资源、软件资源、知识资源等。

利用信息技术，把各个巡天波段的数据经过统一的处理就可以形成一个全波段的模拟天空。根据用户的需求，将相关天区的各种数据返回，就仿佛用户进行

了一次成功的观测，倘若再根据科研要求开发出一批数据处理、数据分析、数据可视化、数据挖掘等工具，就等于拥有了一次观测数据及观测后数据处理的一整套工具。把这些组合起来，就是虚拟天文台[1] [2] 。

虚拟天文台的出现就是为了给天文学家提供一系列强大的工具，把各种数据组织起来，它能够把全球范围内的研究资源无缝透明联结起来而组成一个数据密集型在线天文研究环境。它将 TB 甚至 PB 量级的数据、波长从 γ 射线到射电波段的数十亿个天体的图像库、数据挖掘工具和统计分析工具、数千 PB 量级容量的存储设备和每秒运算次数达到万亿次的超级计算设备、以及各主要天文数据中心之间的高速网络连成一体；使世界各地的天文学家可以快速查询各个 PB 量级的数据库；鼓励了研究团体间的实时合作；允许进行大规模的统计研究，首次使数据库的内容可以和复杂精密的数值模拟结果进行对比。这将极大地改变天文学的研究方法，进而使得天文学取得前所未有的新进展。

1.1.2 国内外研究现状

1.1.2.1 各国虚拟天文台的发展

美国国家科学院在天文与天体物理发展的新十年展望中把国家虚拟天文台列为第一优先发展项目。国家虚拟天文台[3] 的概念一经提出，立即引起天文界的广泛重视。各国也纷纷相继出台了自己的虚拟天文台计划，中国也是其中一员。当前国际主要的虚拟天文台研究项目如图表 1 所示。各国虚拟天文台都面临着共同的需求并寻求着相似的解决方案，从整个天文界的眼光来看，虚拟天文台并不局限于每个项目所属的国家内部，而是建立全球范围内的国际虚拟天文台(The International Virtual Observatory, IVO)。

项目	地区	网址
中国虚拟天文台 (China-VO)	中国	http://www.china-vo.org/
国家虚拟天文台 (NVO)	美国	http://www.us-vo.org/
英国虚拟天文台 (AstroGrid)	英国	http://www.astrogrid.org/
欧洲虚拟天文台 (Euro-VO、AVO)	欧洲	http://www.euro-vo.org/
法国虚拟天文台 (VO-France)	法国	http://www.france-vo.org/
加拿大虚拟天文台 (CVO)	加拿大	http://services.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/cvo/
印度虚拟天文台 (VO-India)	印度	http://vo.iucaa.ernet.in/~voi/
德国天体物理虚拟天文台 (GAVO)	德国	http://www.g-vo.org/
澳大利亚虚拟天文台 (Aus-VO)	澳大利亚	http://www.aus-vo.org/
日本虚拟天文台(JVO)	日本	http://jvo.nao.ac.jp/
俄罗斯虚拟天文台 (RVO)	俄罗斯	http://www.inasan.rssi.ru/eng/rvo/
意大利虚拟天文台 (DRACO)	意大利	http://www.asiat.astro.it/draco/DRACO-home.htm
韩国虚拟天文台 (KVO)	韩国	http://kvo.kao.re.kr/
西班牙虚拟天文台 (SVO)	西班牙	http://www.laeff.inta.es/svo/
亚美尼亚虚拟天文台 (ArVO)	亚美尼亚	http://www.aras.am/ArVO/arvo.htm
匈牙利虚拟天文台 (HVO)	匈牙利	http://www.h-vo.org/

图表1 各国虚拟天文台项目

1.1.2.2 国际虚拟天文台联盟（IVOA）

上一节中提到的各国的虚拟天文台项目，虽然来自不同的国家，有着不同的研究背景和重点，但是他们之间有许多非常重要的共同点：每个项目都在寻求数据密集型天文研究的出路，都在力图挖掘现有以及未来海量天文数据的潜力。从国际天文界的眼光来看，这些项目的共同目标就是建立一个国际虚拟天文台（IVO）。如果把这些项目联合起来，共同迎接虚拟天文台所面临的科学与技术挑战，这对 IVO 的建立是很有好处的。

其中的主要原因是，IVO 必须是一个完整的，不同部分能进行互操作的系统。为了实现不同部分之间的互操作性，就必须对许多问题在国际范围内达成一致和认可。这其中最重要的是数据和接口的标准化，此外需要协调的部分还包括软件包、源代码库、开发工具等。为了 IVO 的最终实现还有一些事情要与官方的政策、资金和安全等问题相关。虽然各国的 VO 项目已就一些互操作性方面的标准达成一致意见，但要取得天文学界更广泛的认可，还需要完成一些更高级的科学和技术演示，同时要公布一个共同的进度安排。

共同进度计划的执行，国际上不同项目间的协调与合作，需要一个机制来实现。正是出于此目的，AstroGrid、AVO和NVO利用 2002 年 6 月在德国Garching 举行“迈向国际虚拟天文台（Towards an International Virtual Observatory）国际会议的机会提出了成立国际虚拟天文台联盟（IVOA）[4] 的倡议，希望利用IVOA把这些国际努力团结在一起。他们的倡议得到了与会代表的支持。IVOA当即表示成立，当时成员数为八个。目前（截止到 2006 年 4 月），IVOA的成员数已经增加到了十六个。

1.2 中国虚拟天文台概述

为了紧跟时代的发展，缩小与发达国家的信息天文学方向的差距，2002 年以中国科学院国家天文台为首在国内天文界首先提出建设“中国虚拟天文台[5]项目”，简称“China-VO”。同年十月，China-VO加入国际虚拟天文台联盟(IVOA)。

1.2.1 China-VO 的研究领域

根据国内实际情况和需求，以及 China-VO 自身的条件和特点，China-VO 的研究开发工作主要在如下五个方面进行：

1. 中国虚拟天文台系统平台的开发（China-VO Platform）：在先进的 IT 技术基础上，按照 IVOA 互操作标准搭建 China-VO 系统平台。
2. 国内外天文研究资源的统一访问（Uniform Access to On-line Astronomical Resources and Services）：研究如何在 China-VO 系统平台上以与 IVOA 标准兼容的方式实现对分布式异构天文研究资源与服务的统一访问。
3. 支持 VO 的项目与观测设施（VO-ready Projects and Facilities）：协助国内外天文学家和工程技术人员，把手中的天文观测项目和观测设施建设为支持 VO 标准的项目和设施。
4. 基于 VO 的天文研究示范（VO-based Astronomical Research Activities）：利用 VO 环境开展天文研究活动，积累 VO 科研经验，带动天文学家使用 VO 资源。
5. 基于 VO 的天文科普教育（VO-based Public Education）：利用 VO 丰富的资源开展天文教育和普及活动。

China-VO 利用以数据网格、XML、WEB Service 等为代表的新兴技术来实现中国的天文数据中主要天文数据的无缝透明访问和集成，对外发布为国际上统一的标准接口，以此来向国际虚拟天文台提供一定数量的数据资源服务，实现与国际 VO 资源的互联共享。天文数据资源的无缝透明访问与互操作是上述其他研究方向的基础，所有的应用服务均在其上开发。因此，数据访问与互操作的

研究又是目前研究解决的重中之重。

1.2.2 中国虚拟天文台的科学目的与意义

1) China-VO 将采用与我国现在正在建造的大天区面积多目标光纤光谱望远镜 (LAMOST[6]) 项目紧密结合的方式, 充分发挥LAMOST 光学光谱数据中心的作用, 建设“VO-oriented LAMOST”, 使光谱数据及其相关处理技术成为 China-VO 的核心与特色。

2) China-VO 将利用先进的网络平台, 方便、快捷、实时地把今后 LAMOST 产出的数据发布, 以便能和国外的数据进行交互、融合。

3) China-VO 将最大程度的集中国内各天文研究机构、IT 研究机构和数学等相关领域的人才资源, 共同努力实现目标。

4) China-VO 将采用开放的运作方式, 与国际上各虚拟天文台计划开展充分的合作, 在 IVOA 中发挥积极作用。

5) China-VO 将利用 IVO 丰富的资源加强教育和科学普及工作, 提高我国公众的科学素质。

作为 e-Science 时代中国天文学研究的新平台和“网关”, China-VO 把国内的资源共享到国际, 把国际的资源输入到国内, 逐步实现与国际天文数据的互操作、透明访问。最终, China-VO 的用户将可以通过使用互联网的方式方便地获取国际范围内的天文数据, 可以对分布在各地的数据进行大规模多波段的分析和挖掘, 并利用 China-VO 的可视化系统将结果以 2D 图像的形式显示出来。作为国际虚拟天文台的成员, China-VO 将与国际上各个虚拟天文台进行紧密合作, 在 IVOA 中发挥积极作用。

1.3 虚拟天文台数据访问服务 (VO-DAS) 的研究意义

中国虚拟天文台 (China-VO) 进行的虚拟天文台数据访问服务系统 (VO-DAS) 研究是在 2005 年开始的。此系统着眼于解决天文数据的数据雪崩以及各个天文数据保存以及管理方式的不统一等问题而开发的。

统一访问这些分布存放的异构数据这个课题,一直是各国虚拟天文台的研究重点。NVO、JVO、CDS 等都进行了很多尝试。到目前为止,虽然已经有若干数据访问方案提出,还没有一种能完全满足天文学家的需求的服务,它们要么因为所选择的网络技术而限制了其功能,要么因为为专门的项目服务而限制了应用范围和互操作性。一个理想的天文数据访问系统应该具备以下特点:

- 能够访问分布保存的异构天文数据库;
- 支持自动发现数据资源;
- 支持特定空间区域的数据访问;
- 支持海量数据访问;
- 支持天文数据的交叉认证;
- 支持星表数据,图像数据和光谱数据的访问;
- 具有标准的数据访问接口;
- 具有良好的互操作性,可以和任何符合 IVOA 规范的 VO 应用程序互联。

依据这样的需求,我们设计了中国虚拟天文台自己的数据访问服务(VO-DAS)。此系统建立在Globus Toolkits 4[7] (GT4)之上。它能够实现异地异构数据库的访问,数据资源的自动发现,海量数据的访问与查询,多种数据格式的访问,天文数据的交叉认证[8] 等。此外,它的对外接口简单实用,可以针对不同需求的天文数据用户发展出多种网格应用服务。

2 虚拟天文台数据标准

2.1 天文数据的特点

因为虚拟天文台数据访问服务（VO-DAS）系统主要针对天文数据的处理而开发，而虚拟天文台数据访问服务注册系统（VO-DAS Registry）系统更是针对天文数据的元数据进行处理。所以，这里先对天文数据的特点进行一些介绍，然后，本章将对VO-DAS Registry系统涉及到的三个关键技术和概念进行一些介绍。这三个技术是XML[10]、VOTable[11]和ADQL[12]。

在介绍这些技术之前，先介绍一下天文数据的特点：

1). 天文数据绝大部分是开放数据。天文界的传统是所有的观测数据在一年后向公众开放，使得观测者有时间进行数据分析和发表早期结果，也使其他天文学家可以有机会使用这些数据。国际上的许多大型天文观测项目的观测数据都会及时在互联网上公布，这为数据共享提供了良好的基础，在各类学科中是独一无二的。

2). 天文数据的数据量非常大。现有的以及即将实施的天文项目每天都会产生数 GB 甚至 TB 的数据，天文数据中心的存贮容量已经达到数 TB，并开始向 PB 扩展。

3). 天文数据有比较好的归档，并提供互联网服务。当前，世界上已经有多家天文数据中心在天文数据归档方面做了大量的工作，并取得了很好的应用，甚至各类天文文献也是在线网络服务的。

4). 天文数据的格式多种多样。天文数据的内容主要有星表、星图、光谱等，数据的格式则各种各样，其内部格式都依不同的天文观测项目而变化。

5). 天文数据是全波段的数据。从 γ 射线、X 射线、紫外、光学、红外到射电波段都有观测项目在进行。这些数据是高度相关的，需要在高维参数空间内进行研究。

6). 新的观测数据总能促进全新的现象或规律的发现。

2.2 可扩展标记语言 (XML)

2.2.1 简介

Web 使我们能够与任何地方的任何人通讯。广泛接受的标准（这对完全挖掘 Web 的潜力是至关重要的）允许 Web 在多种交互操作的技术层上通讯。要想设定智能搜索、数据交换、自适应表示和个人化的标准，Internet 必须设置信息理解标准（表示数据的通用方式），以便软件能够更好地搜索、移动、显示和处理上下文中隐藏的信息。由于 HTML 是一种描述如何表示 Web 页的格式，尽管提供了用于显示的丰富工具，但它并没有提供任何基于标准的管理数据的方式。正如数年前用于显示的 HTML 标准扩展了 Internet 一样，数据标准亦将扩展 Internet。数据标准将是商业交易、自动协作和数据共享的工具。

可扩展标记语言 (Extensible Markup Language) 简称 XML，是 Web 应用的一种新技术，是万维网联盟 [9] (W3C) 制定的标准。XML 简化了网络中数据交换和表示，使得代码、数据和表示可以分离，可以作为数据交换的标准格式。就这点而言，XML 被称为智能数据文档。

W3C 对 XML 进行了如下描述：XML，描述了一类被称为 XML 文档的数据对象，并部分描述了处理它们的计算机程序的行为。XML 是 SGML (Standard Generalized Markup Language [ISO 8879]) 的一个应用实例或一种受限形式。从结构上说，XML 文档顺从 SGML 文档标准。

XML 提供可以编码各种情况（从简单至复杂）的内容、语义和架构的数据标准，可被用于标注以下对象：

- 普通文档。
- 诸如约会记录或采购定单之类的结构记录。

- 具有数据和方法的对象，如 Java 对象或 ActiveX 控件的持续型表单。
- 数据记录，如查询的结果集。
- 有关 Web 站点的元内容，如“频道定义格式 (CDF)”。
- 图形表示，如应用程序的用户界面。
- 标准架构实体和类型。
- Web 上的信息与用户之间的所有链接。

总而言之，XML 是一种元标注语言，该语言提供一种描述结构数据的格式。这有助于更精确地声明内容，方便跨越多种平台的更有意义的搜索结果。此外，XML 将起用新一代的基于 Web 的数据查询和处理应用程序。XML 在采用简单、柔性的标准化格式表达和在应用间交换数据方面迈出了一大步。HTML 提供了显示数据的通用方法；XML 则提供了直接在数据上工作的通用方法。XML 的威力在于将用户界面和结构化数据相分离，允许不同来源数据的无缝集成和对同一数据的多种处理。从数据描述语言的角度看，XML 是灵活的、可扩展的、有良好的结构和约束；从数据处理的角度看，它足够简单且易于阅读，几乎和 HTML 一样易于学习，同时又易于被应用程序处理，因此，XML 必将带来下一代网络应用技术的革命。

2.2.2 XML 文档

XML 是一种基于文本的格式，在许多方面类似于 HTML，后者是专为存储和传输数据而设计的。XML 源是由 XML 元素组成的，每个 XML 元素包括一个开始标记(<title>)、一个结束标记(</title>)以及两个标记之间的信息（称为内容）。就像 HTML 一样，XML 文档保存利用标记注释的文本。然而，与 HTML 不同的是，XML 允许无限的标记集，各标记集并不表示如何显示，而是表示其含义。例如，可以将 XML 元素标记为价格、订单编号或名称。由文档的作者确定使用何种数据以及哪种标记名称最合适。

标记是对了文档存储格式和逻辑结构的描述。在形式上，标记有以下各种可能项：注释、引用、字符数据段、起始标记、结束标记、空元素、文档类型声明（DTD）和序言。

每个 XML 文档都有一个逻辑结构和物理结构。从物理角度来看，文档由实体单元组成，一个实体也可以在其他文档的实体中被引用。一个文档以一个根元素或文档实体来开始。从逻辑上讲，文档由声明（declaration）、元素（element）、注释（comment）、字符引用（character reference）和处理说明（processing instruction）组成。这些组成部分在文档的标记中必须明确规定。物理结构从另一角度来规范 XML 文档。文档的起始标记和结束标记对数据进行结构化组织，并确定了元素的范围和相互之间的关系。

在 XML 文档中，除标记之外就是字符数据。字符数据段中除了不能有结束标志"]]"外，可以包括任意字符序列。例如，标记化的文本可以直接放在字符数据段中而不会被解释为结构化的标记。我们可以在字符数据段中包含一段 HTML 代码，也可以在字符数据段中添加编程语言代码。字符数据段为 XML 的后续处理提供了编程接口。

XML 文件也可以对一个处理 XML 数据的应用提供处理信息，即处理说明，其格式为："<?目标应用名 instructions?>"。例如，<?xml version="1.0"?>就是一种处理指令。解析器通过它得知该文档应当遵循 XML 1.0 标准。XML 文件可以对不同应用提供不同处理说明。

XML 文件总有一个"序言（prolog）"声明该文件为 XML 文档，例如（<?xml version="1.0"?>）。该序言也可以包含附加信息，如：

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

Prolog 的属性包括：版本、编码以及是否独立，后者指明该文档是否引用一个外部实体或外部数据类型。另外还可以直接包含本来在 DTD 中声明的实体定义和规范。

从上面的介绍，我们可以看出，XML 有以下几个特点：

- XML 文档是纯文本，因此可用从文本编辑器直到可视化开发环境的任何工具创建和编辑，程序可以更简单，而在需要存储大量数据的场合，一个数据库的 XML 前端就可以满足需要，因此 XML 提供了从小配置文件到公司级数据仓库的可扩展性。
- 基于内容的数据标识，因而可被不同程序用于不同用途。
- 可格式化。在显示很重要的时候，XSL 可指定如何显示数据。由于数据和显示是分离的，甚至可以为同一数据指定不同的样式表用于不同输出，并很容易在将来使用新的格式。
- 具有很强的链接能力，可以定义双向链接、多目标链接、扩展联结和两个文档间的链接。
- 易于处理。XML 对格式的定义更为严格，并具有层次结构，处理起来更加容易。它是与厂商无关的标准，可以任选一个解析器来处理。

2.2.3 架构 (XML Schema)

在 XML 中，“文档类型声明(DTD)”可以伴随文档，且文档类型定义实质上定义文档的规则，例如哪些元素可被显示以及元素之间的结构关系。当接收应用程序没有传入数据的内部描述时，DTD 将帮助验证该数据。然而，在 XML 中，DTD 是可选的。

与 DTD 一起发送的数据称为有效 XML。此时，XML 语法分析程序应对照 DTD 中定义的规则校正传入的数据，以确保该数据被正确构造。没有与 DTD 一起发送的数据称为结构良好的 XML。

对于有效 XML 和结构良好的 XML，XML 编码的数据是自描述的，这是由于描述标记和数据是混合的。XML 使用开放的、灵活的格式，在任何需要交换和传送信息的地方均可使用 XML，这使得 XML 功能更加强大。

DTD 本身规定了有限的 XML 文档规则，这与 XML 的初衷有所违背：XML

采用可扩展的标记希望能够表示各种数据，但 DTD 又规定了有限的 XML 文档规则，限制了 XML 文档。架构（Schema）是 XML 文档规则的正式规格，即元素名称，架构表示文档中允许哪些元素以及允许哪些组合。

XML Schema 是 W3C 制订的用于定义和验证 XML 语法的标准。一个 XML Schema 通常描述下列信息：

- 一系列合法的 XML 标签和属性
- 标签的排列顺序
- 合法的数值

使用架构，作者正好可以定义在文档中允许什么样的元素名，以及在每个元素中允许什么样的子元素、属性和关系。作者可从其他架构中引入片断，以及通过继承来扩展类型。于是，在保留了词汇树结构简单性的同时，还允许元素间的复杂关系。

作者可以发明他们自己的架构，或者共享别的作者创建的架构。读者可以检查架构参考以检验已经收到的文档的类型是否正确。他们还可以使用架构中的信息来自动检验文档的结构。

2.2.4 范例

一个典型的架构例子如图表 2 所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://nvoss.org/VOResource"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:element name="resource">
<xs:complexType>
<xs:sequence>
<xs:element name="title" type="xs:string" />
<xs:element name="referenceURL" type="xs:anyURI" minOccurs="0"/>
<xs:element name="type"
minOccurs="0" maxOccurs="unbounded">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="Archive" />
<xs:enumeration value="Catalog" />
<xs:enumeration value="Organisation" />
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
<xs:attribute name="created" type="xs:dateTime" />
</xs:complexType>
</xs:element>
</xs:schema>
```

图表2 一个典型的 XML 架构例子

2.2.5 XML 数据处理

XML 不仅规定了如何表示和显示数据，还提供了标准的 API 供处理 XML 数据，这也就是我们称之为智能数据或数据标准的原因。

SAX[11]（The Simple API for XML，SAX）是基于事件的XML分析API，功能比较简单。这一API是事件驱动的，又称"顺序访问"协议。每当它看到一个新的XML标记（或遇到一个错误，或想告诉用户什么事时）就用一个SAX解析器注册你的句柄，激活用户的回调方法。

DOM [14] 定义了分析程序应当显露的标准命令集，使您能够在程序中访问 HTML 和 XML 文档内容。支持 DOM 的 XML 分析程序取出 XML 文档中的数据，并通过一组可以对它编程的对象来显露它。DOM将一个XML文档转换成用户程序中的一个对象集合。然后用户可以任意处理对象模型。这一机制也称为"随机访问"协议，因为用户可以在任何时间访问数据的任何一部分，然后修

改、删除或插入新数据。DOM的特点是功能强大，但分析时间长，占用资源多。

SAX 提供了处理 XML 文档的快速、低内存的另一种方法。在使用 DOM 分析 XML 文件时，它在内存中建立了完整的文档树。相比而言，SAX 将遍历文档，并将新元素的开始或结束等通知分析事件的调用应用程序。使用 SAX 的一个最佳功能是分析长文档。例如，用 SAX 分析器，应用程序可以监视发生的事件，只将文档中必要的部分读入内存。

2.3 VOTable 天文表列数据的封装

VOTable是天文界为满足数据互操作性的需要而制定的基于XML语言的表数据集合的格式规范，用做灵活交换传输和存储表列结构数据的，特别是作为天文表列数据的标准数据格式。VOTable格式前身是Astrores格式，VOTable格式结构接近于FITS[15] 二进制表(FITS Binary Table)的格式标准。VOTable采用XML格式标准使得应用很容易验证VOTable的有效性，并能很容易地转换（利用XSLT转换器）成其他数据格式。

VOTable 的一个重要的特点是提供了对表的元数据的描述定义。VOTable 不仅仅包括表数据，而且还包括对描述这些表数据的元数据。元数据包括对表自身的描述和对表中各属性列的描述，在对属性列的描述中还引入了描述列的具体天文语义的 UCD，使得计算机和人都可以理解每一列数据表达的具体天文意义。

VOTable 格式能够满足大数据量网格计算的需要。VOTable 格式允许数据和元数据分离存储，即在文档中仅仅包含指向远程数据的链接(而不包含数据本身)和元数据信息。在网格计算中，处理器可以根据这些元数据信息为接受远程数据做好准备，或者是组织第三方传输或并行传输远程的数据。当我们在网格分布式计算环境中运用大数据量的表列数据时，数据流在异地的处理器间传输，并被过滤、和缓存，如果需要在数据表的头部提供数据的行数的话（比如 FITS）会带来很大的麻烦和困难，必须要把整个数据表输入流缓存在存储器中，计算有多少行，然后把数据再次转换成数据流，要知道，在网格中最缺乏的不是处理器，而是大的缓冲存储器，而 VOTable 不需要在文件头提供数据的大小。另外，由于

VOTable 可能是由其他的处理器动态生成或存储在一个暂时的存储器中，VOTable 可以提供一个参数描述经过多久远程数据将不再存在。在网络上的天文数据有可能需要获取授权，所以 VOTable 也提供描述了密码和身份确认信息的参数。

VOTable 是灵活高效的数据存储格式。VOTable 的数据部分可以选择采用三种不同的存储格式，包括 TABLEDATA, FITS, BINARY。TABLEDATA 是一个纯 XML 的格式，所以能采用常规的 XML 工具对小数据表高效处理；FITS Binary Table 是天文学家最熟悉的格式，VOTable 能作为数据传输和存储的标准格式，也能作为只存储元数据的标准格式。VOTable 可以被用来将整个 FITS 文件封装，或重新对元数据进行重新编码；很不幸的是最困难的是流化 FITS 文件，因为 FITS 头需要数据的长度值，并且需要确定变长数组的最大长度；BINARY 格式即采用二进制存储数据，采用这种格式节省存储空间并且容易编写程序，在存储空间的利用上甚至强于 FITS(FITS 文档的一个记录长度必须是 2880 字节，会浪费空间)，并且读写 BINARY 数据也不需要专门的 FITS 的程序库。

2.4 ADQL 数据查询标准

ADQL (Astronomical Data Query Language)被设计用来描述查询表列结构天文数据要求的查询语言，它基于结构化查询语言(SQL)，采用 XML 文档格式，它实际上是一种 XML 格式的 SQL 语言。ADQL 设计的重要目的是作为天文统一的数据查询语言，实现天文数据的分布式查询，目前 90%天文数据库是关系型数据库，而且大部分天文表列格式的数据是由各种关系型数据库管理的，ADQL 最重要的是屏蔽了各关系数据库平台实现 SQL 语言标准的不同（许多数据库管理系统对 SQL 语言做了不同的扩展），它可以对存储在各种数据库中的天文数据的查询进行统一的描述，为实现多数据库平台下的天文数据统一查询访问和互操作提供了解决途径。而且为了弥补 SQL 缺乏描述空间区域以及时空坐标等天文专用的语句，满足方便简洁描述区域查询和坐标变换的要求，ADQL 还提供了一

些描述区域查询、交叉认证和时空坐标系的语法，并且计划在将来做更大的扩展和完善，以满足查询表列数据以外的复杂数据如图像数据和光谱数据的要求，比如增加新的数据类型和操作等等。目前 ADQL 已经成功的应用到 SkyQuery 中和很多项目中，我们也采用 ADQL 作为星表查询服务的输入查询语言。

3 虚拟天文台注册系统（VO Registry）

构建 VO-DAS Registry 系统，最重要的工作就是对虚拟天文台注册系统（VO Registry）的研究。而 VO Registry 都是在 IVOA 的相关协议和标准的框架下构建的。所以本章重点对 IVOA 的相关协议进行了较多的阐述，然后对英国虚拟天文台的 AstroGrid Registry 和美国国家虚拟天文台的 STScI/JHU NVO Registry 这两个本 VO-DAS Registry 系统已经成功实现连接的 VO Registry 进行了介绍。

3.1 国际虚拟天文台联盟制订的相关标准

IVOA在Registry方面制定了如下标准：关于元数据的协议[16] 和关于 Registry访问接口[17] 的协议。注册标准旨在为如何满足天文学家在任何VO环境中对任意地点任意资源的发现和使用的需求提供解决方案。

3.1.1 资源元数据标准

为了更好的管理和使用数据以及数据服务，就必须对元数据[18] 进行很好的管理，虚拟天文台资源元数据标准就是着眼于这一点而制订的。

虚拟天文台资源元数据标准定义了一个体系结构，在这个体系结构的顶层，只需要少量的信息说明资源的存在和其出处。在下面的层次中，元数据越来越丰富和复杂，对数据内容、查询语法、访问协议、使用规则进行说明。这样的体系结构可以让天文信息服务能方便地加入到 VO 中。

在虚拟天文台资源元数据标准模型中：

- 资源（Resource）在 VO 中是一个非常宽泛的词汇，几乎所有东西都可称为资源。一个资源是一个或多个服务或其它资源的集合。资源有一些共同的元数据特征，比如发布者、创立者、标识、类型等。资源和服务都由元数据来描述。
- 组织（Organization）是一个特定类型的资源，由人员组成。

- 服务（Service）是 VO 中能被用户调用来执行某些操作的实体。
- 查询服务（Query Service）支持查询/响应操作。用户向服务提出查询，服务向用户反馈一些信息。
- 注册（Registry）是一个查询服务，其响应是一个关于其它服务的结构化的描述。

IVOA 给出的资源服务元数据标准主要的内容如下：

资源元数据

● 标识元数据

标题（*Title*）：

数据类型：string

定义：资源的名称

名称(*shortname*)

数据类型：string

定义：名称缩写，限于 16 个字符

● 履历元数据

发布者（*Publisher*）

数据类型：string

定义：发布资源的实体，它要对资源的可用性负责

发布者标识（*PublisherID*）

数据类型：URI

定义：资源发布者的标识，可以是一个网络链接

创建者（*Creator*）

数据类型：string

定义：资源主要内容的创建实体，可以是个人或者组织

主题（*Subject*）

数据类型：string, list

定义：对资源的主题、天体类型或其他属性进行描述的一系列关键词

描述 (*Description*)

数据类型: string, free text

定义: 对资源内容的一个概括

贡献者 (*Contributor*)

数据类型: string

定义: 对资源内容有贡献的实体

日期 (*Date*)

数据类型: string

定义: 在资源的生命期中一个重要的日期, 比如创建或者发布日期

版本 (*Version*)

数据类型: string

定义: 与资源创建或者发布相关的一个标号

资源参考 (*ReferenceURL*)

数据类型: URL

定义: 一个指向资源额外信息的 URL 地址

联系方式 (*Contact*)

数据类型: string, e-mail address

定义: 资源负责人, 包括联系人的姓名和电子邮件地址

● 内容元数据

主题 (*Subject*)

描述 (*Description*)

来源 (*Source*)

ReferenceURL

关系 (*Relationship*)

关系标识符 (*RelationshipID*)

类型 (*Type*)

数据类型: string, list

定义: 资源内容的种类或类型, 可以是文档 (*Archive*)、星表 (*Catalog*) 等其中的一个或几个。

范畴 (*Coverage*)

数据类型: string

定义: 资源内容的涵盖范畴, 包括空间覆盖 (*Coverage.Spatial*)、相关区域覆盖 (*Coverage.RegionOfRegard*)、光谱覆盖 (*Coverage.Spectral*)、时间覆盖 (*Coverage.Temporal*)、深度覆盖 (*Coverage.Depth*)、目标密度 (*Coverage.ObjectDensity*)、目标计数 (*Coverage.ObjectCount*) 等子项。

内容等级 (*ContentLevel*)

数据类型: string, list

定义: 内容等级描述, 指出资源的潜在用户, 可以是公众 (*General*)、小学教育 (*Elementary Education*)、初级中学教育 (*Middle School Education*)、高级中学教育 (*Secondary Education*)、社区学院 (*Community College*)、大学 (*University*)、业余研究 (*Amateur*)、信息教育 (*Informal Education*) 中的一项或几项。

格式 (*Format*)

数据类型: string, list

定义: 资源所提供信息的物理或者数字显示方式, 比如文件格式 (*FITS*, *ASCII*, *VOTable*, *GIF*, ...)、存储方式 (*CDROM*、*DVD*、在线、录像带、出版物, ...) 等

权限 (*Rights*)

数据类型: string

定义: 资源的所属及访问权限

机构 (*Facility*)

数据类型: string

定义：获取数据的天文台和研究机构

设备 (*Instrument*)

数据类型：string

定义：收集数据所用设备

服务元数据

● 接口元数据

服务接口 *URL* (*ServiceInterfaceURL*)

数据类型：URL

定义：一个指向服务接口说明文档的 URL

服务基准 *URL* (*ServiceBaseURL*)

数据类型：URL

定义：用户激活服务的 URL 的基本部分

HTTP 服务接口返回结果 (*ServiceHTTPResults*)

数据类型：(MIME type)

定义：服务返回结果的 MIME 类型

● 功能元数据

服务标准 *URI* (*ServiceStandardURI*)

数据类型：URI

定义：标识一个服务标准的 URI

服务标准 *URL* (*ServiceStandardURL*)

数据类型：URL

定义：一个指向服务标准描述的 URL

服务最大搜索范围 (*ServiceMSR*)

数据类型：float, decimal degrees

定义：服务提供者或服务附加的最大搜索范围，以度的形式规定最大的搜索半径

图表 3以斯隆数字巡天获取的星表为例，展示各个数据段的内容：

Identity metadata	
Title	Sloan Digital Sky Survey
ShortName	SDSS
Identifier	ivo://stsci.edu/mast/sdss
Curation metadata	
Publisher	Space Telescope Science Institute/MAST
PublisherID	ivo://stsci.edu/mast
Creator	Sloan Digital Sky Survey Consortium
Creator.Logo	http://archive.stsci.edu/images/sdss_logo.gif
Contributor	Sloan Digital Sky Survey Consortium
Date	2003-02-01
Version	SDSS EDR
ReferenceURL	http://archive.stsci.edu/sdss/index.html
Contact.Name	Archive Branch, Space Telescope Science Institute
Contact.Email	archive@stsci.edu
General content metadata	
Subject	galaxies, quasars, stars, CCD photometry,spectroscopy, redshift, sky surveys
Description	The Sloan Digital Sky Survey is using a dedicated 2.5 m telescope and a large format CCD camera to obtain images of over 10,000 square degrees of high Galactic latitude sky in five broad bands (u', g', r', i' and z', centered at 3540, 4770, 6230, 7630, and 9130 Å, respectively). Medium resolution spectra will be obtained for approximately 10⁶ galaxies and 100,000 quasars. The early data release (EDR), on June 2001, includes searchable catalogs of images and spectra, images for display and scientific purpose in both 2-D FITS and JPEG formats, and spectra in both 1-D FITS and GIF formats. The EDR covers about 460 square degrees of sky. The next data releases will occur every 18 months or so.

虚拟天文台注册系统（VO Registry）

Source	2002AJ....123..485S	
Type	Survey, Catalog, EPOResource	
ContentLevel	Research	
Relationship	mirror-of	
RelationshipID	ivo://sdss.org/sdss/edr	
Collection and service content metadata		
Facility	Apache Point Observatory, Sloan 2.5-m Telescope	
Instrument	Five-band clocked CCD camera	
Coverage.Spatial	polygon (FK5, 145.17, 1.25, 235.9, 1.25, 235.9, -1.25, 145.17, 1.25) or polygon (FK5, 250.71, 66.29, 267.0, 66.29, 267.0,52.15, 250.71, 66.29) or polygon (FK5, 350.43, 1.17, 360.0, 1.17,360.0, -1.25, 350.43, -1.25) or polygon (FK5, 0.0, 1.17, 56.37, 1.17, 56.37, -1.25, 0.0, -1.25)	
Coverage.RegionOfRegard	0.0001	
Coverage.Spectral	Optical	
Coverage.Spectral.Bandpass	u', g', r', i', z'	
Coverage.Spectral.MinimumWavelength	400.e-9	
Coverage.Spectral.MaximumWavelength	850.e-9	
Coverage.Temporal.StartTime	1999-12-25	
Coverage.Temporal.StopTime	2001-07-15	
Coverage.Depth	3.e-6	
Coverage.ObjectDensity	6.e4	
Coverage.ObjectCount	2.e7	
Coverage.SkyFraction	0.01	
Resolution.Spatial	0.00028	
Resolution.Spectral	5000	
Resolution.Temporal	120	
UCD	Not Provided	
Format	text/xml	
Rights	Public	

Data quality metadata	
DataQuality	A
Uncertainty.Photometric	3.e-7
Uncertainty.Spatial	0.00003
Uncertainty.Spectral	1.e-11
Uncertainty.Temporal	0.1
Service metadata	
Service.InterfaceURL	http://archive.stsci.edu/sdss/catalog.html
Service.BaseURL	http://archive.stsci.edu/cgi-bin/sdss/catalog
Service.HTTPResultsMimeType	text/xml
Service.StandardID	ivo://ivoa.net/Services/ConeSearch
Service.StandardURL	ivo://www.ivoa.net/Documents/REC/ConeSearch.html
Service.MaxSearchRadius	0.2
Service.MaxReturnRecords	5000

图表3 斯隆数字巡天的资源元数据

3.1.2 Registry 访问接口协议

制定此协议的目的是保证 Registry 之间以及 Registry 与程序之间的互操作性。因为数据本身的分布式性质，所以 VO 环境中必定有着多个 Registry 存在，这样，保证 Registry 之间的通信以及互操作性就成为非常重要的问题。此协议描述了一个标准的接口。凡是按照实现了此接口的 Registry 系统，其他程序、用户以及 Registry 系统就能够使用相同的方式与此系统交互。同样的，凡是符合此标准的客户端都能够从兼容的 Registry 中发现和获取资源。

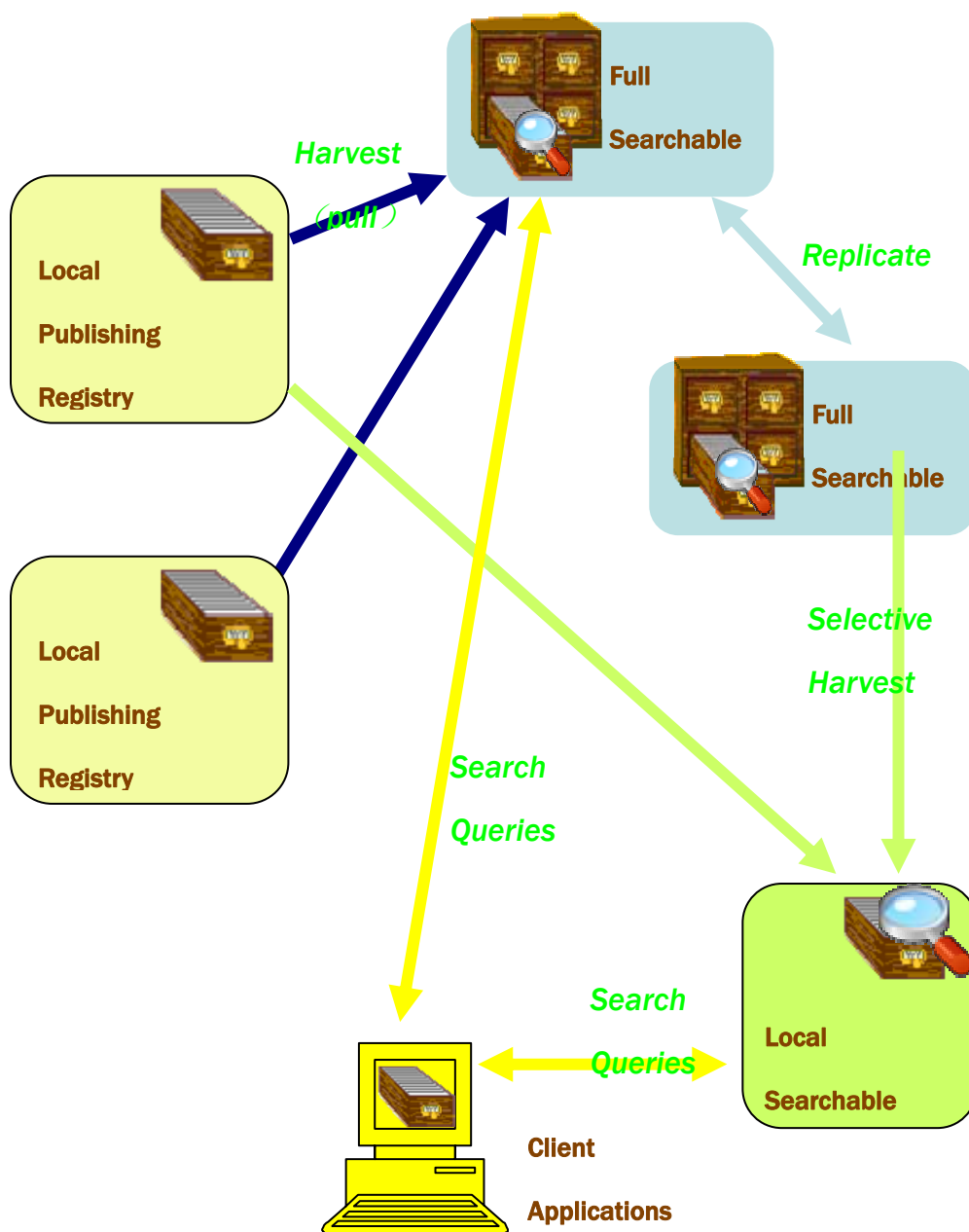
根据 IVOA 制订的最新版本的 Registry 访问接口标准，Registry 可以看作是一个容器，里面存放的是关于数据或者服务资源的描述信息。Registry 存放的描述信息就是这些数据或者服务的元数据（Metadata）。而这些数据或者服务本身则是分布于世界各地的异构系统。

IVOA 把 Registry 根据功能分成以下类别：首先是可搜索 Registry 和发布式 Registry；然后是全局 Registry 和局域 Registry。

可搜索 Registry 是指可供用户或者客户端程序对 Registry 中的元数据进行搜索查询服务的 Registry 系统。这一类 Registry 系统通常保存着大量的数据，所以往往基于大型的数据库系统。可搜索 Registry 系统通过网络获取多个的分布于各地的数据或者服务资源的元数据，而这个获取的过程，我们称为收获 (harvesting)。此外，收获也指两个 Registry 之间进行的同步操作。相比之下，发布式 Registry 系统结构比较简单，它只需要把它拥有的数据或者服务资源以可搜索 Registry 系统能够收获的方式发布出去就可以了。它通常不和用户或者客户端程序进行交互。而它所拥有的数据或者服务资源往往由一个或者少数几个提供者所发布。

全局 Registry 则是试图涵盖 VO 中已知的所有数据或者服务资源。全局 Registry 往往同时是可搜索的，即全局性可搜索 Registry。基于 IVOA 的标准，会有多个这种 Registry 系统存在，它们往往由大型的虚拟天文台项目所维护。我们 VO-DAS 的 Registry 系统就是一个全局性可搜索的 Registry 系统。局域 Registry 系统顾名思义，只包含部分的数据或者服务资源信息。事实上，所有的发布式 Registry 系统都是局域的。

各类 Registry 之间的关系如图表 4 所示。



图表4 各种 Registry 的交互

Registry 系统存在的意义就在于方便用户或者客户端程序对数据或者服务资源的访问。只要 Registry 和客户端程序的接口符合 IVOA 制订的 Registry 访问接口标准，它们之间的交互就能够正常的进行。

Registry 访问接口事实上包含两个独立的部分：获取接口和查询接口。获取

接口是 Registry 之间进行交互使用的,而查询接口才是供客户程序查询使用。查询接口必须支持来自客户端或者其他 Registry 的 ADQL 形式的查询以及关键字查询。而对于查询的返回结果则是符合查询条件的一系列资源 (VOResource) 的描述信息。Registry 系统可根据实际需要实现任一个或者全部的接口。VO-DAS 当前采用英国虚拟天文台项目 AstroGrid 开发的 Registry 以及美国国家虚拟天文台的 STScI/JHU NVO Registry。这两个 Registry 已经实现全部的接口。

3.2 各个 VO Registry 介绍

目前,很多VO项目都开发了自己的Registry系统,比较成功的VO Registry 系统列表如图表 5所示。

Registry 名	所有者	网址	可搜索否?	可发布否?
ESAVO Full Harvestable VO Resource Registry	ESAVO	http://esavo.esa.int/registry/	是	是
AstroGrid Registry	AstroGrid	http://software.astrogrid.org/components/registry/overview.html		
STScI/JHU NVO Registry	NVO	http://nvo.stsci.edu/VORegistry/index.aspx	是	是
Caltech Carnivore	NVO	http://nvo.caltech.edu:8080/carnivore/	是	是
NCSA Registration Portal	NVO	http://nvo.ncsa.uiuc.edu/nvoregistration.html	否	是

图表5 几个 VO Registry 的基本信息

下面将以更直观的方式列出各个VO的VO Registry系统的位置,可见现在各种符合IVOA标准的Registry产品很多,但大多数处于实验品阶段。这里展示的地图使用的是Weasl[19] 的地图服务。Weasl是AstroGrid提供的能够非常直观的展示IVOA的各种服务资源的一个网站。

欧洲各国 VO 的 Registry 系统:



图表6 欧洲各国 VO 的 Registry 系统

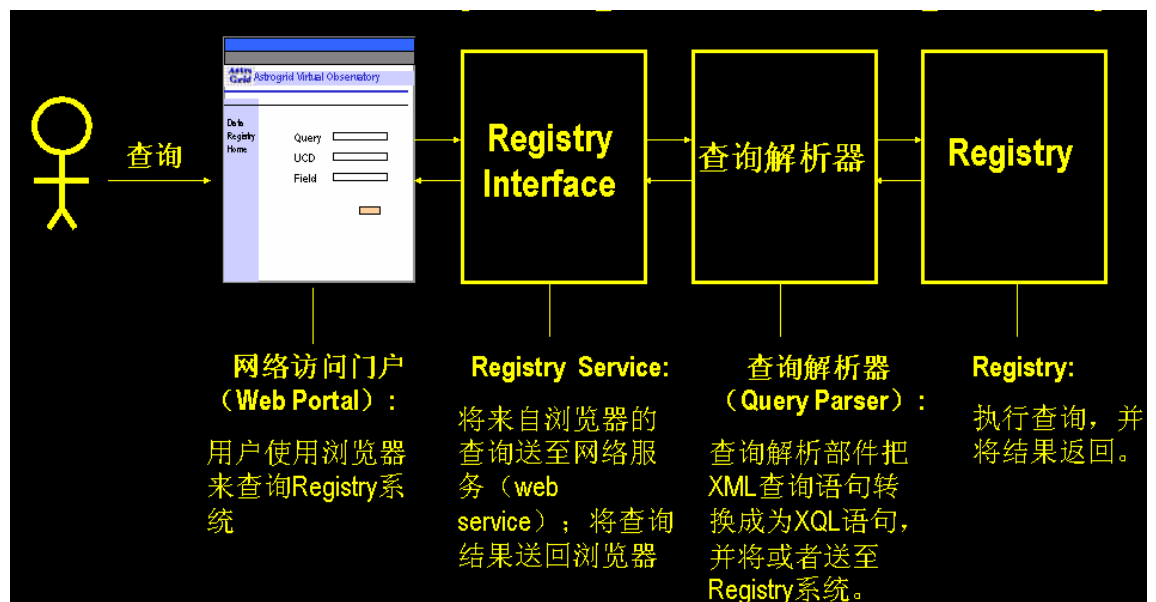
亚洲和美洲各个 VO 的 Registry 系统，特别地，将 China-VO 目前安装运行的 AstroGrid Registry 系统的相关信息列出来：



图表7 亚洲、美洲各虚拟天文台的 Registry 系统

3.2.1 AstroGrid Registry

AstroGrid Registry是英国虚拟天文台开发的天文网格AstroGrid项目的组成部分。它提供了友好的查询界面，其系统模型如图表 8 AstroGrid Registry系统模型所示：



图表8 AstroGrid Registry 系统模型

对于已注册的资源，用户可以在网页中进行浏览、关键字查询和ADQL查询。

图表 9是用户进行浏览资源得到的结果，而图表 10则是用户使用china作为关键词查询的结果，可以看到已经注册的DAS的资源。点击Edit可看到资源的XML描述，内容应类似图表 11所示。

Registry Browser

Find IVORNs including:

List

Title	Type	AuthorityID	
China VO publish registry 0.1	vg:Registry	naoc.ac.cn.pub	org.
astrogrid.cat.test	vg:Registry	astrogrid.cat.test	org.
FileManager Service	vr:Service	naoc.ac.cn.pub	filem
FileStore Service	vr:Service	naoc.ac.cn.pub	files
Community Authority	vg:Authority	naoc.ac.cn.pub	null?
Policy Manager Community Service	vr:Service	naoc.ac.cn.pub	org.
Policy Service Community	vr:Service	naoc.ac.cn.pub	org.
Security Manager	vr:Service	naoc.ac.cn.pub	org.
Security Service	vr:Service	naoc.ac.cn.pub	org.

图表9 AstroGrid Registry 资源浏览页面

Query Registry

Keyword search. Place words seperated by spaces.

Keywords:
 Require all words: ☐ All words

**If no errors the Query results will be below:
 Only a max of 25 entries will be shown:**

AuthorityID	ResourceKey	View XML
naoc.ac.cn.pub	org.astrogrid.registry.RegistryService	View,Edit
naoc.ac.cn.pub	organisation	View,Edit
naoc.ac.cn.pub	null?!	View,Edit
naoc.ac.cn.pub	DAS1	View,Edit
naoc.ac.cn.pub	DAS2	View,Edit
naoc.ac.cn.pub	DAS3	View,Edit
naoc.ac.cn.pub	DAS4	View,Edit

图表10 AstroGrid Resgisty 资源搜索页面

而对于 AstroGrid 的管理者，包括资源注册、资源收获、删除等常见操作也都可以使用网页方便的进行。

```

<ri:Resource created="2007-05-17T02:14:19" status="active" updated="2007-05-17T02:14:19" xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0" xmlns:str="urn:astrogrid:schema:StoreResources:v1" xmlns:vg="http://www.ivoa.net/xml/VORegistry/v1.0" xmlns:vr="http://www.ivoa.net/xml/VOResource/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="vg:Registry">
  <title>China VO publish registry1.0</title>
  <identifier>ivo://naoc.ac.cn.pub/DAS1</identifier>
  <curation>
    <publisher>LY</publisher>
    <contact>
      <name>Yong LU</name>
      <email>luyong@bao.ac.cn</email>
    </contact>
  </curation>
  <content>
    <subject>VO_DAS_1</subject>
    <description>DAS1</description>
    <referenceURL>http://192.168.3.150:7070/dataresource/votable_3galx_delData_forGD.xml</referenceURL>
    <type>Registry</type>
  </content>
  <capability standardID="ivo://ivoa.net/std/Registry" xsi:type="vg:Harvest">
    <interface role="std" xsi:type="vg:OAIHTTP">
      <accessURL>http://192.168.3.150:7070/wsrf/services/ogsadai/DataService</accessURL>
    </interface>
    <maxRecords>200</maxRecords>
  </capability>
  <full>false</full>
  <managedAuthority>naoc.ac.cn.pub</managedAuthority>
</ri:Resource>

```

图表11 在 AstroGrid Registry 注册了的 DAS 资源例子

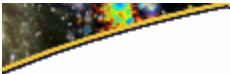
3.2.2 NVO Registry

美国国家虚拟天文台由多个机构和大学组成，他们开发了多个Registry系统，包括DAS现在使用的STScI/JHU NVO Registry以及Carnivore[20] 等系统，后者和AstroGrid Registry相似，发布可被使用者安装在自己的服务器上的产品。此类产品适合于用户需大规模注册自己的资源的场合。而STScI/JHU NVO Registry提供了通过网页注册资源和丰富的通过程序查询资源的服务，适用于用户只有少数资源需注册的场合。本节只针对STScI/JHU NVO Registry进行介绍。

与AstroGrid Registry相似，用户也可以方便的使用网页进行资源浏览、注册、查询等功能。图表 12是浏览资源出现的页面，可见，此Registry现有资源 14653

个。按照资源类型（ResourceType）进行了分类。

相应的，STScI/JHU NVO Registry 提供了根据资源类型进行资源查询的程序接口。只要使用字符串 ResourceType like ‘OTHER’，STScI/JHU NVO Registry 就会返回以 OTHER 类型的资源，事实上，本文中的 VO-DAS Registry 系统就是使用这种方式查询 STScI/JHU NVO Registry 的。



Registry Contents by ResourceType

Click on a resource Type to get a browsable listing for that type.

ResourceType	Resource Count	Description
AUTHORITY	36	a naming authority; an assertion of control represented by an authority identifier
CEASERVICETYPE	14	Astrogrid Common Execution Architecture S
CONE	448	Simple Cone Search (IVOA Standard)
DATACOLLECTION	32	A logical grouping of data which, in general accessible datasets.(A dataset is a collecti with a that is normally accessible as a sing
HTTP-GET	2	Simple Web Form
N/A	3	Not Applicable (not recommended)
ORGANISATION	20	A publisher of other resources (At a high le university, observatory, or government age be a specific scientific project, mission, or
OTHER	4	Other (not recommended)
REGISTRY	13	A service that provides access to descripti
SERVICE	19	Astrogrid Resource Spec for Web Service
STAR	64	Simple Image Access Service (IACS Stand

图表12 STScI/JHU NVO Registry 资源浏览页面

比AstroGrid Registry更方便的地方是STScI/JHU NVO Registry的资源注册，用户只需要根据范例进行填表操作即可，如图表 13所示：



Searchable Registry

Home	Contents	Search	Publish	Delete
------	----------	--------	---------	--------

National Virtual Observatory : NVO Registry

To copy an existing record enter the Identifier here:

Do not forget to enter your password and a NEW identifier for inserting the record.

Title?			
Shortname?		Identifier?	
ContactName?		ContactEmail?	
Creator?		Publisher?	
Contributor?		Subject?	
ResourceType	SERVICE		
Description?			
Related Resources			
Type?		Instrument?	
Date?	5/19/2007 4:50:05 AM	Version?	
ReferenceURL?		ServiceURL?	
CoverageSpatial?	Could not find a part of the path "c:\inetpub\nvo\voregistry\region.xml".		CoverageTemporal?
RegionOfRegard?	0	CoverageSpectral?	
ContentLevel?		Facility?	

图表13 STScI/JHU NVO Registry 资源注册页面

4 虚拟天文台数据访问服务（VO-DAS）注册系统

4.1 VO-DAS 系统以及 VO-DAS Registry 在其中的地位和功能

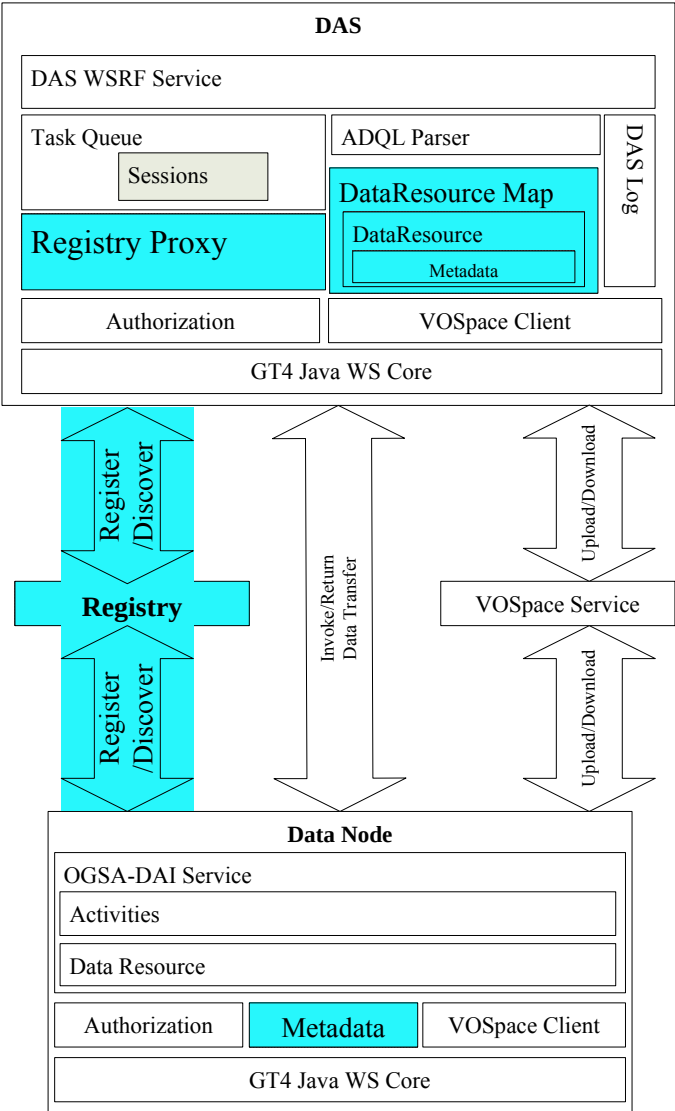
DataNode 是虚拟天文台数据访问服务（VO-DAS）系统中提供数据资源的服务，它是建立在 OGSA-DAI WSRF 中间件的基础上的网格服务。OGSA-DAI 很好地封装了各种不同类型的数据库，统一了它们的访问接口。不仅是数据库，OGSA-DAI 还支持将一个文件集作为数据库进行访问。DataNode 在安装和配置成功以后应当注册到一个 VO Registry 服务中。

VO-DAS Registry 系统是 VO-DAS 的重要组成部分，它基于 IVOA 制订的 Registry 系统的标准，为 VO-DAS 系统发现、获取数据资源及数据资源的元数据服务。并且 VO-DAS 处理查询请求的时候，基于 VO-DAS Registry 系统，可以提供给查询用户可用的数据资源是哪些，减少数据查询的盲目性，提高查询的效率。

如图表 14所示，VO-DAS系统分为两个部分，DataNode和VO-DAS。DataNode用来封装VO-DAS中的所有天文数据库。所有的DataNode都需按照IVOA的Resource Metadata规范注册到VO Registry中去，VO-DAS Registry可架构在不同的VO Registry之上，图中的Registry指的就是VO Registry，可以是任何实现了IVOA相关协议的Registry系统。目前我们已经成功地将本系统架构在了AstroGrid Registry和STScI/JHU NVO Registry之上。也就是说VO-DAS Registry系统已经实现了与AstroGrid Registry以及STScI/JHU NVO Registry相连接的接口。

VO-DAS 模块中，Registry Proxy 在系统启动之初建立 DataResource Map，并且在系统运行过程中自动更新它。在这个过程中，Registry Proxy 使用 Registry

访问接口标准去访问 AstroGrid 或者 STScI/JHU NVO Registry 提供的 Registry 服务。在二者的 Registry 服务中,包含了所有 DataNode 的元数据信息。Registry Proxy 模块会读取这些信息并且生成对应的 Metadata 对象,保存在 DataResource Map 中。



图表14 VO-DAS Registry 在 VO-DAS 中的功能

在 VO-DAS 中,任何一个查询请求都是用 ADQL 语言来描述的,ADQL 语言需要 ADQL Parser 模块来进行解析并生成执行计划。在这个过程中,ADQL Parser 就需要查询 DataResource Map 来获取 ADQL 语句的 From 子句中的表存在

于哪个 `DataNode` 中。这里可能会有两种例外情况：第一，这个表不存在，则返回查询失败；第二这个表存在多个 `DataNode` 中，则使用查询到的第一个 `DataNode` 上的表。

查询语句经过解析后，会生成一个待执行的执行计划，在合适的时候，连接 `DataNode`，执行此计划并且把执行结果保存到指定的存储空间。

根据上面的描述可知，VO-DAS Registry 系统在整个 VO-DAS 中起着沟通 `DataNode` 和 VO-DAS 两大模块的作用。Registry Proxy 是 VO-DAS 中读取 VO Registry 服务的模块，并且建立和更新 `DataResource Map` 模块。`DataResource Map` 模块则是生成执行计划必须的。

4.2 系统设计

根据上面提到过的 IVOA 关于 Registry 和 Metadata 的标准，为了服务于 VO-DAS 系统，我们设计的 VO-DAS Registry 系统要能够从保存所有 `DataNode` 的 Metadata 信息的 VO Registry 获取并解析它们的 Metadata 信息。按照 IVOA 的 Metadata 标准，关于 `DataNode` 的 Metadata 信息被分为以下几个部分，每部分都包含多个描述 `DataNode` 的元数据信息：

标识元数据 (Identity metadata)

履历元数据 (Curation metadata)

内容元数据 (General content metadata)

范畴元数据 (Collection metadata)

服务元数据 (Service metadata)

数据和质量元数据 (Data and metadata quality)

对应上述 6 个部分，我们设计的 `MetaData` 对象包含 6 个对象，用来描述它代表的 `DataNode` 的元数据，如图表 16 所示。对象名分别是：`MDItemIdentity`、`MDItemCuration`、`MDItemGeneralContent`、`MDItemCollection`、`MDItemService`、

MDItemDataQuality。其中MDItemCollection中还包含子对象MDItemResolution、MDItemCoverage，MDItemService中还包含子对象MDItemServiceInterface、MDItemServiceCapability。因篇幅所限，图表 15中仅列出其中两个对象作为说明。

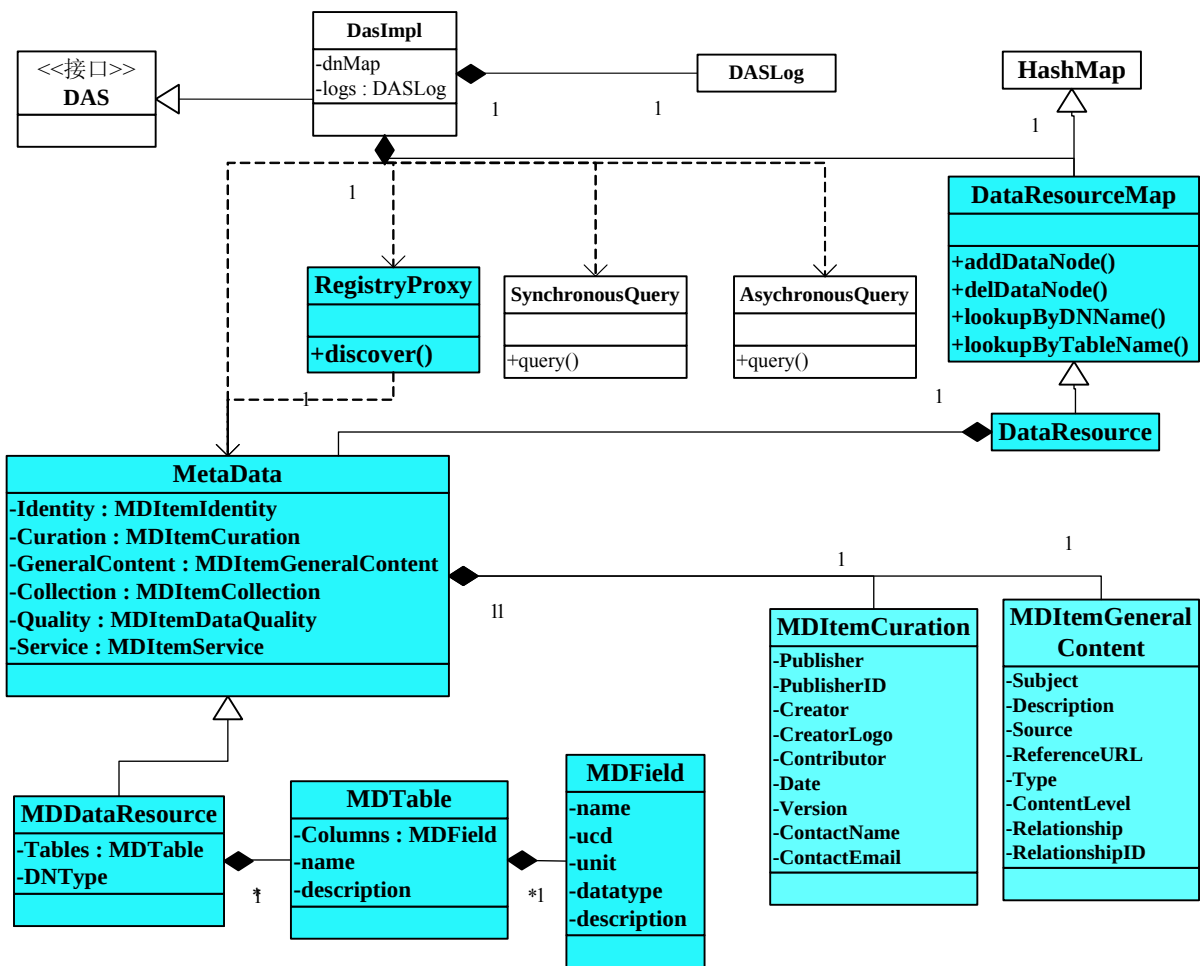
MDDataResource 对象继承了 Metadata 对象。它又包含多个 MDTable 对象，对应这个 DataNode 包含的表格的元数据；而 MDTable 对象除了包含表格的名字、描述信息之外，还包含多个 MDField 对象，对应于表格的各个列的元数据信息。

系统启动以后，VO-DAS Registry的工作流程如图表 17所示。首先被调用的方法是RegistryProxy对象中的discover方法。discover方法会读取VO Registry中保存的所有关于DataNode的Metadata信息。对于每一个DataNode，我们将它的元数据信息解析之后写入到一个MDDataResource中去。而为了这个DataNode中各个表格的元数据信息，则需要解析出从每个DataNode的元数据信息中的InterfaceURL。InterfaceURL指向的是关于此DataNode包含数据的Metadata文件，得到并且解析此文件，就可以获取此DataNode包含的表格的所有元数据信息，包括每个表格的描述和表名，以及它所包含的所有列的元数据信息。然后将每个列的信息写入MDField，再将每个表的信息写入MDTable，在此之后，将写好的这些MDTable写入对应的MDDataResource。最后，将此MDDataResource写入到一个DataResource对象中，并将此对象插入到DataResourceMap中。到此为止，对一个DataNode的元数据就分析和保存完毕。循环对每个DataNode进行这个步骤之后，就完成了对DataResourceMap对象的构建。

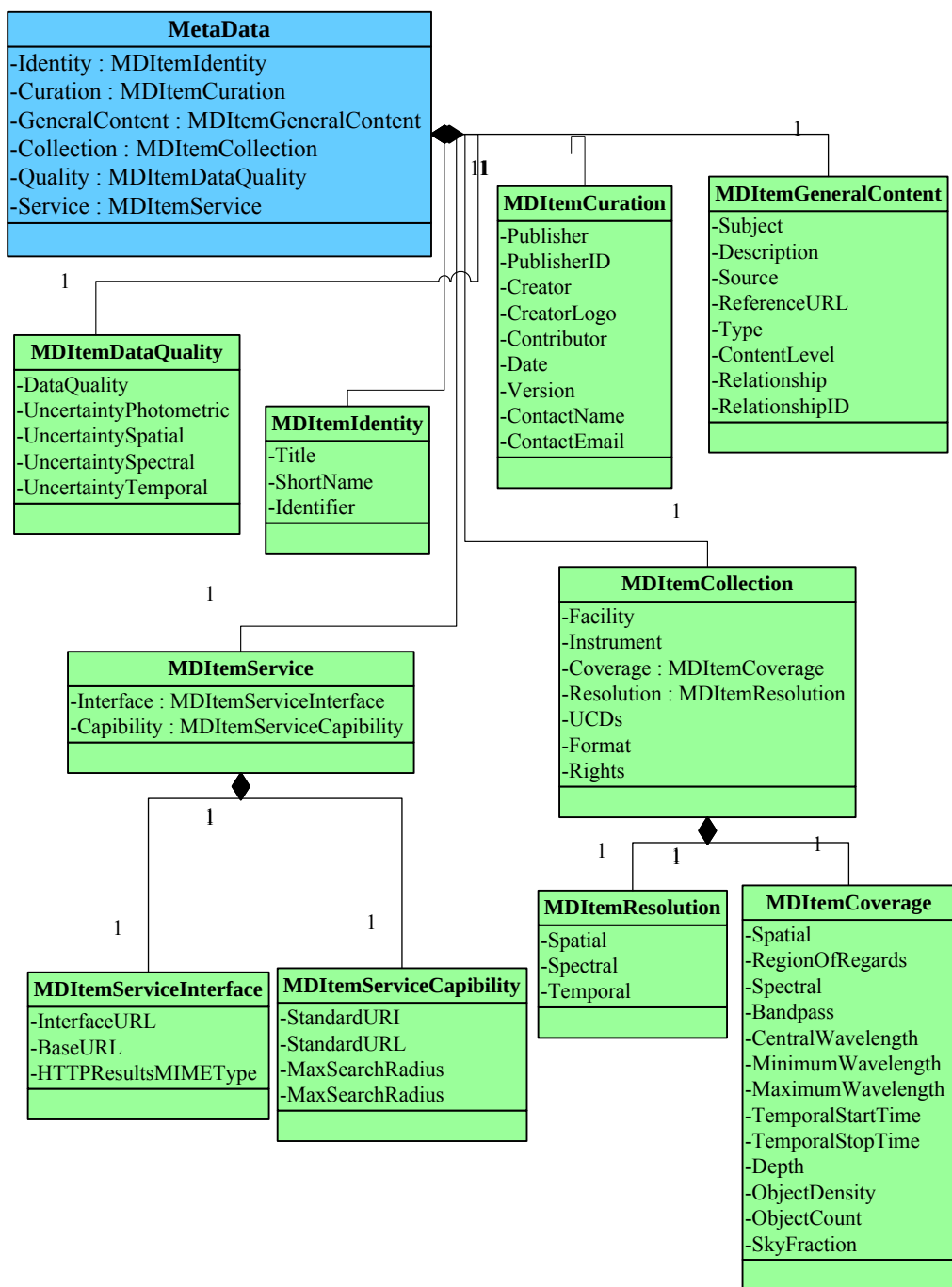
为了方便添加和维护我们将 DataResourceMap 对象设计为一个 HashMap。这样做的目的是更方便地管理 DataResource。在系统运行过程中，当发现一个新的 DataNode 时，VO-DAS Registry 系统会获取其元数据并通过上述过程将其加入到 DataResourceMap 中；反之，当 VO-DAS Registry 系统发现 DataResourceMap 中保存的 DataResource 对应的 DataNode 无法访问的时候，就会主动把它从 DataResourceMap 中删除。

本系统与DAS的其他模块、VO Registry等的交互可参考图表 18所示。其中，DAS的Engine模块调用VO-DAS Registry的discover方法后，本系统开始运行。

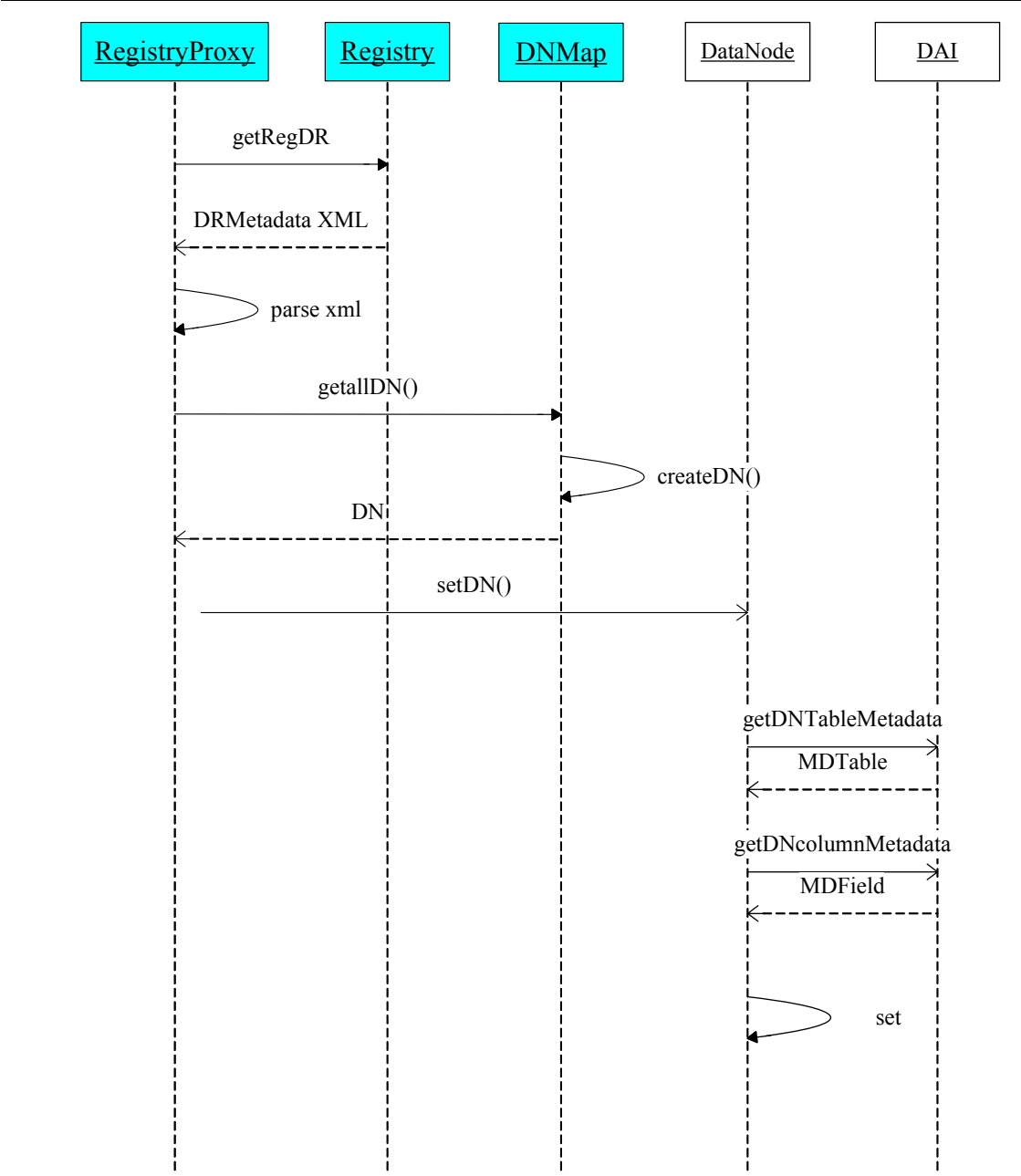
File1、File2 是InterfaceURL指向的文件，保存了对应的DataNode中数据（如星表数据）的原数据信息。



图表15 VO-DAS Registry 系统设计(1)

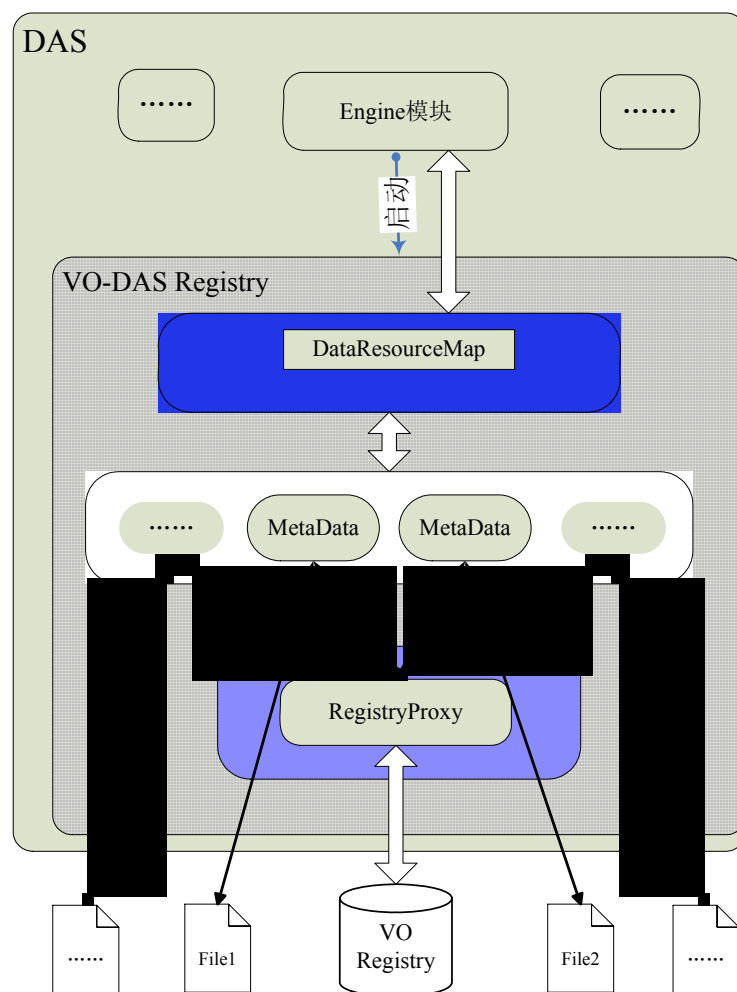


图表16 VO-DAS Registry 系统设计(2)



图表17 VO-DAS Registry 流程示意图

本系统采用Java[21] 编码实现，本节所述的所有模块都已经完成。



图表 18 VO-DAS Registry 与其它模块交互图

4.3 与 VO Registry 的连接

4.3.1 与 AstroGrid Registry 的连接

AstroGrid Registry提供了程序调用的接口。程序可以使用WebService方便地进行关键字查询、XQuery[22] 查询等操作。返回的结果则是符合条件的资源XML描述。一个典型的XQuery查询字符串如下：

```
"declare namespace xsi = 'http://www.w3.org/2001/XMLSchema-instance';
declare namespace vr = 'http://www.ivoa.net/xml/VOResource/v0.10';
declare namespace vor = 'http://www.ivoa.net/xml/RegistryInterface/v0.1';
//vor:Resource[@status='active' and ((near(vr:title,'naoc') or
near(vr:shortName,'naoc') or near(vr:content/vr:subject,'naoc') or
near(vr:content/vr:description,'naoc') ))] "]"
```

此 XQuery 查询元数据的 title、shourname、subject、description 四个字段中只要有一个包含 naoc 这个关键词的资源。

本系统调用 AstroGrid Registry 主要使用了 org.astrogrid.registry.client.query.RegistryService 这个包。图表 19 中展示的是使用 RegistryService 类的 xquerySearch 方法对 Registry 进行 Xquery 查询的语句。此方法返回结果是个 Document 对象，以下就可以使用程序来处理返回结果了。

```
RegistryService rs = RegistryDelegateFactory.createQuery(new URL("http://archive.lamost.org/registry-cvo-pub/services/RegistryQuery"));
System.out.println("got the registry service try calling loadregistry");
String xquery = "declare namespace xsi = 'http://www.w3.org/2001/XMLSchema-instance'; declare namespace vr = 'http://www.ivoa.net/xml/VOResource/v0.10'; declare namespace vor = 'http://www.ivoa.net/xml/RegistryInterface/v0.1'; //vor:Resource[@status='active' and ((near(vr:title,'naoc') or near(vr:shortName,'naoc') or near(vr:content/vr:subject,'naoc') or near(vr:content/vr:description,'naoc') ))] ";
Document doc = rs.xquerySearch(xquery);
```

图表19 与 AstroGrid Registry 连接程序的一段代码

4.3.2 与 STScI/JHU NVO Registry 的连接

STScI/JHU NVO Registry 提供了 HTTP GET 和 SOAP 两种程序接口。前者使用起来非常方便，而后者功能强大。

一个典型的 HTTP GET 字串是这样的：

<http://nvo.stsci.edu/VORegistry/registry.asmx/QueryResource?predicate=ResourceType%20like%20'OTHER'>

此字串查询 Registry 中所有资源类型 (ResourceType) 是 OTHER 类型的资源。返回结果是一个 XML 文档。

本系统调用HTTP GET主要依靠GetLink方法，主要内容如图表 20所示。系统将查询字符串以及生成临时文件的位置发送给此方法，此方法根据查询字符串组合成HTTP GET字串，进行Registry查询。和AstroGrid Registry相连接不同的是，这里查询的结果保存在一个临时文件中。以下使用程序来处理此文件即可。

```
public class GetLink{

    GetLink() {    }

    public boolean writefile(String key,String filename)
    {
        FileOutputStream fos=null;
        try
        {
            String temp="http://nvo.stsci.edu/V0Registry/registry.asmx/QueryResource?predicate="+key;
            URL url = new URL(temp);
            try
            {
                System.out.println("Try to write files");
                InputStream is = url.openStream();
                InputStreamReader isr=new InputStreamReader(is);
                BufferedReader bf=new BufferedReader(isr);
                //File f=new File("d:\\library.xml");
                File f=new File(filename);
                fos=new FileOutputStream(f,false);
                //dos=new DataOutputStream(fos);
                String str;
                while((str=bf.readLine())!=null){
                    //dos.writeBytes(str);
                    str=str+"\n";
                    fos.write(str.getBytes());
                }
                return true;
            } catch (IOException ex1)
            {
                ex1.printStackTrace();
                System.out.println("Write Error!");
                return false;
            }
        }
    }
}
```

图表20 与 STScI/JHU NVO Registry 连接的一段代码

5 其他相关工作

5.1 图像和光谱的联合查询

5.1.1 简单图像访问协议（SIA）与简单光谱访问协议（SSA）

SIA(Simple Image Access)和 SSA(Simple Spectral Access)协议都是数据获取层（Data Access Layer）的协议。简单地说，SIA 协议是在指定天区搜索并获取图像数据的协议，而 SSA 则是在指定天区搜索、获取光谱的协议。

SIA、SSA 协议的工作过程都可以分为两个部分：定位图像和获取图像。

定位图像中，它们都使用 HTTP GET 方法，访问 CGI 服务。客户端必须提供的三个参数是天区的中心位置（POS）、天区的尺寸（SIZE）和返回图像/光谱的格式（FORMAT）。POS 是用逗号分割的、以度为单位的赤经和赤纬；SIZE 则是用逗号分割的角距离；FORMAT 则可以是多种格式，如 image/fits, image/jpeg。而 HTTP 返回的则是 VOTable 格式的图像/光谱，在这个 VOTable 中，每行代表一个符合要求的图像。在此行的列 VOX:Image_AccessReference 中是这个图像的 URL 地址，客户端可以据此获取图像。按照 SIA、SSA 协议，服务器端传回的图片可能是本来就存在于服务器中的，也可由服务器临时生成。

下面以NASA[23] 的SkyView[24] 提供的服务为例，可以对SIA协议有较为直观的印象：

此 CGI 服务的地址是：

<http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?survey=digitized&>

我们进行一个查询，指定天区的中心是赤经 99 度，赤纬 10 度，大小是 0.1 度。查询语句，也就是浏览器的地址栏输入的语句应为：

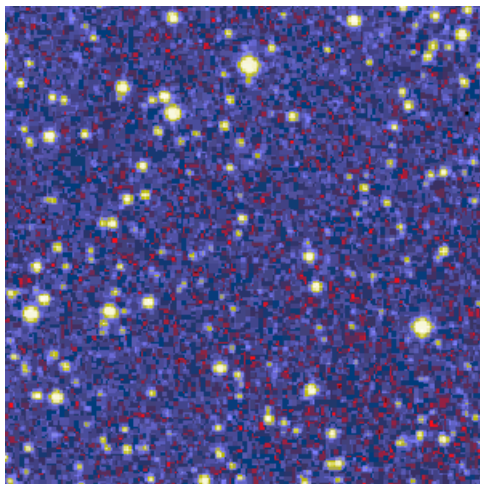
<http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?survey=digitized&POS=99,10&SIZ=0.1&FORMAT=ALL>

这样，我们查询的结果就是一个VOTable的XML文档，在浏览器中显示部分如图表 21所示。

```
+ <PARAM value="0" name="INPUT:small"></PARAM>
- <TABLE>
  <FIELD name="Survey" ucd="VOX:Image_Title" datatype="char" arraysize="1"/>
  <FIELD name="Ra" ucd="POS_EQ_RA_MAIN" datatype="double"/>
  <FIELD name="Dec" ucd="POS_EQ_DEC_MAIN" datatype="double"/>
  <FIELD name="Dim" ucd="VOX:Image_Naxes" datatype="int"/>
  <FIELD name="Size" ucd="VOX:Image_Naxis" datatype="int" arraysize="1"/>
  <FIELD name="Scale" ucd="VOX:Image_Scale" datatype="double" arraysize="1"/>
  <FIELD name="Format" ucd="VOX:Image_Format" datatype="char" arraysize="1"/>
  <FIELD name="PixFlags" ucd="VOX:Image_PixFlags" datatype="char" arraysize="1"/>
  <FIELD name="URL" ucd="VOX:Image_AccessReference" datatype="char" arraysize="1"/>
  <FIELD name="NBytes" ucd="VOX:Image_FileSize" datatype="int"/>
  <FIELD name="LogicalName" ucd="VOX:LogicalName" datatype="char" arraysize="1"/>
- <DATA>
- <TABLEDATA>
+ <TR></TR>
- <TR>
  <TD>Digitized Sky Survey 1</TD>
  <TD>99.</TD>
  <TD>10.</TD>
  <TD>2</TD>
  <TD>300.300</TD>
  <TD>-0.000333333333333333,0.000333333333333333</TD>
  <TD>image/gif</TD>
  <TD>F</TD>
- <TD>
  http://skyview.gsfc.nasa.gov/cgi-bin/pskcall?SURVEY=Digitized%20Sky%20Survey&VCOORD=99.%2C10.
  </TD>
  <TD>90000</TD>
  <TD>1</TD>
```

图表21 一个使用 SIA 服务的查询结果（1）

其中<TABLEDATA>中一个包含URL的<TD>标签就是这个图像的URL地址，这是一个GIF格式的文件。从这个地址，我们可以获得图像如图表 22所示。



图表 22 一个使用 SIA 服务的查询结果 (2)

5.1.2 设计

我们进行联合查询中的一个重要概念就是虚拟表 (Virtual Table)。虚拟表是为了使用 ADQL 语言构造包括图像和光谱查询而提出的。ADQL 使用者可以把图像和光谱库看作如同星表般存储方式存在于远端的服务器上, 根据虚拟表的一列或者多列, 使用 ADQL 语言, 用户可以通过特性指定查询的图像和光谱, 并且限定返回图像和光谱的哪些属性。

虚拟表包含两部分, 输入属性和输出属性, 使用输入属性, ADQL 语言来限定返回怎样的图像和光谱, 而返回的图像和光谱的属性可以通过选择的输出属性来限定, 当然, 为了方便, 也可以选择同时输出图像或者光谱的输入属性。对应于 SIA、SSA 协议。输入属性对应于协议中规定的可出现于具有查询功能的 URL 中的属性, 输出属性对应于协议规定的表示查询结果的 VOTABLE 的每一行的各列的值。输入属性和输出属性大部分是不同的, 但有些是重叠的, 在重叠的情况下, 按照协议规定, 使用不同的名称来区分。

一个光谱和图像联合查询的例子如下:

```
Select
image.image.title,image.INST_ID,image.Format,spectrum.DataSet_
type,spectrum.title,spectrum.curation_collection,image.image_A
ccessReference,spectrum.Access_reference,spectrum.Access_forma
t
From image,spectrum
Where .image.POS=(90,10) AND image.SIZE=0.1 AND
spectrum.POS=(80,11) AND spectrum.SIZE=0.05
```

此 ADQL 语句在由 POS 和 SIZE 指定的天区分别进行图像和光谱查询。使用 SIA、SSA 协议，可以实现此查询。

5.2 海量星表数据处理工具的自动化

5.2.1 海量天文数据处理工具的背景和简介

天文数据主要包括星表、星图、光谱、文献资料等，其中星表是包含天体信息的数据表格，是天文学家最常用到的天文数据。将不同波段的星表、特别是巡天项目产生的大型星表进行交叉认证，一直是进行很多天文学研究、尤其是数据挖掘、统计分析研究的瓶颈。

星表是包含天体信息的数据表格，是天文学家最常用到的天文数据。星表大小差别悬殊，小到几十个目标，大到上亿甚至上十亿个目标，而且近年来随着巡天项目增加和扩大得到的大星表越来越多。星表包括 ReadMe 文件和数据表文件。其中 ReadMe 文件以标准格式提供星表的有关信息，如包含哪些文件，并提供数据表文件中每一列的位置、名称、单位等描述（即表结构）。一个星表对应着一个数据库(即 Database)，一个星表可能包含多个数据表。星表的一个数据表对应着数据库的一个表(即 table)。

北京天文数据中心[25] (BADC)作为中国目前最大的天文数据中心，已经建立了自己的天文数据库和网络数据服务系统，并通过国际交换和网络下载积累了大量适合我国天文研究需要的天文数据。它积累的天文数据库是虚拟天文台的重要要素之一，也是其研究开发的基础和实验床。天文星表数据库作为国家天文数

据中心数据库的重要部分，收集和整理重要的和最新的星表数据集，为国内外天文学家和其他用户提供星表查询服务和星表获得服务。

为了实现对星表的高效管理和查询，我们设计实现一个功能完善的天文星表数据库管理系统。天文星表数据库的主要数据来源于斯特拉斯堡天文数据中心[26] (CDS) 负责收集整理天文星表，也包括其他大型巡天项目和其它星表数据。高丹等[27] 讨论了如何基于北京天文数据中心实现了星表的自动入库工具，把CDS、SDSS[28]、2MASS[29] 等其他数据中心的数据自动入库到北京天文数据中心。此后，又实现了多个具有不同功能的天文数据处理工具，包括：自动入库工具、交叉证认工具等。

5.2.2 整合和将海量星表数据处理工具自动化的意义

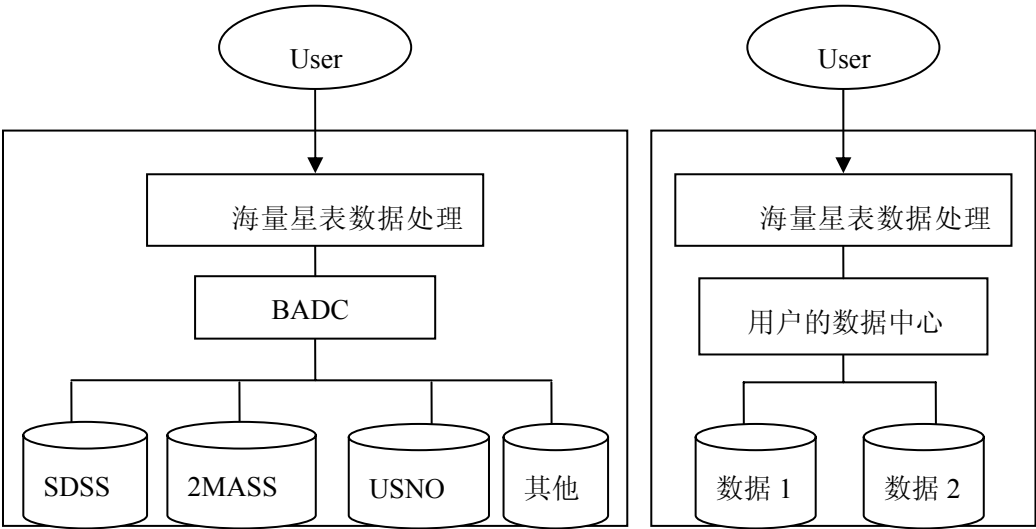
我们的这项工作是在上述工作的基础上，把各个工具进行整合，使得各个工具能够批处理地、多任务地进行工作，且将此工具发布成为一个能够基于任何支持 SQL 语言的数据库系统的、面向使用者的一项服务。这项工作很大程度地方便了交叉证认等工作的进行。

在进行这项工作之前，这些工具都是独立的程序，每个程序使用的时候都需要用户输入很多的参数，很容易出错，易用性方面不尽人意。而且当需要这些程序共同完成一个工作的时候：例如，先将两个星表入库，然后建立 HTM 索引，再进行交叉证认以及提取参数表，最后再将数据返回给用户，就需要用户分别进行多次参数输入，如星表入库、建立索引就分别需要两次，并且每次工作都必须等待上一步工作完成之后，用户来用命令行执行。这样，不仅需要用户反复查询当前工作是否完成，造成工作的低效，而且用户输入参数本身也是一项繁琐而易出错的工作。

现在各个步骤可以使用参数文件方便地进行编辑，原来需要输入的参数列表，现在可以在功能强大的编辑器中，根据我们提供的模版，编辑这些参数，可以很大的提高用户的效率，且减少了出错的可能性；而将各个步骤的灵活组合，更使得完成复杂功能对于用户而言，只需要一次编辑各步骤的参数文件，无论每

一步需要进行多少次，都只是需要编辑此步骤对应的一个参数文件，然后运行即可。

现在此系统可以架构在任何支持SQL语言的数据库上。如图表 23所示，此系统架构在（BADC）上，系统已经多次使用于数据挖掘工作的数据处理工作，运行正常。此系统也可以方便地移植到任何数据库系统上。当用户只是需要拿自己的一些小星表自身进行交叉证认，或者拿自己的小星表和此系统基于的数据库中的星表进行交叉证认的时候，可以上传自己的星表，使用本人的系统提供的服务。而当用户有着大量的星表需要进行交叉证认，或者需要上传大型星表的时候，如果使用别人提供的此服务，可能会在网络传输上花费较多的时间，更方便的办法是自己建立一个数据库，将本文描述的系统架构到自己的数据库上再使用。这对存储着大量数据的各大天文数据中心有着很大的意义。



图表23 海浪星表数据处理工具的架构

综上所述，可见整合前和整合后的对比如下：

整合前：

- 仅仅在 BADC 上运行
- 各个工具彼此独立

- 用户需要使用命令行，输入很长的参数表，易出错
- 每次只能执行单个操作
- 易用性差，效率低
- 用户上手难

整合后：

- 可以架构在任何支持 SQL 的数据库
- 使用参数文件使得用户编辑参数更方便
- 提供了参数文件模版，用户更容易理解和使用
- 各个独立的步骤可以灵活组合
- 批处理地连续完成各个功能，不需要用户干预
- 用户学习很容易

5.2.3 设计

整合这些工具的基本思路就是使用批处理的思想。整合这些工具使用了 Linux Shell编程和Vi[30] 的自动处理功能，所以这些程序运行于Linux操作系统。首先用户根据需要进行的操作编辑批处理文件steps_crosswork，编辑此文件的时候可以参考我们提供的模板steps_crosswork_template。还是假设用户需要执行包括将两个星表都入库和建立索引，再将两个星表进行交叉认证，然后进行参数提取和数据传回。则steps_crosswork文件应类似如下：

```
./step1 step1_file_putdata
./step2 step2_file_HTMindex
./step3_7 step3_file_crossmath7
./step4 step4_file_sqldata
./step5 step5_file_getdata
```

这里，step1,step2 等可以被看作命令，具备执行相应的步骤的功能，而step1_file_putdata 则是 step1 对应的参数文件。注意到，第一步和第二步都事实上对两个星表进行操作，但是这里不需要体现。只需要在在相应的参数文件中体现。

举例而言，对 step1 对应的参数文件 step1_file_putdata 可以是：

```
GLMC_l350.tbl ReadMe test glmc
GLMC_l348.tbl ReadMe test glmc
```

这里 GLMC_l350.tbl、GLMC_l348.tbl 就是用户提供的星表，其他几个则是一些参数。

看 step1 文件，则是：

```
vi -s script1 $1
sh $1
vi -s script1_end $1
```

而我们看 script1 文件，则是

```
:%g/^$/norm dd
:s/^/\./putdata1 /
:wq
```

script1_end 文件则是：

```
:%s/\./putdata1 //
:wq
```

我们能够注意到，step1 文件对它的参数文件，也就是 step1_file_putdata，首先用 script1 脚本，使用 Vi 进行了处理（vi -s script1 \$1），然后运行了处理后的脚本文件，最后使用 script1_end 脚本，仍对 step1_file_putdata 进行了处理。

第一次使用 Vi 处理的过程就是将 step1_file_putdata 文件中的空行去掉（这样做是为了防止用户错误的输入了空行），然后在每个行首加上./putdata1，保存退出。这时 step1_file_putdata 就是：

```
./putdata1 GLMC_l350.tbl ReadMe test glmc
./putdata1 GLMC_l348.tbl ReadMe test glmc
```

然后在 step1 中，就是执行这个被修改了的 step1_file_putdata 文件。我们看到，事实上就是使用 putdata1 命令，参数就是最初的 step1_file_putdata 文件中提供的参数。

看一下 putdata1 文件，我们在这里把此文件称为第一步的功能文件

```
java      -cp      /home/gaodan/intodb/      Dbreadme.Putdata
/data0/gaodan/${4}/${1} /data0/gaodan/${4}/${2} $3
```

这样，在 putdata1 中，GLMC_l350.tbl ReadMe test glmc 这四个参数分别代替了 \${1}、\${2}、\${3}、\${4}。然后执行了这个 java 命令行程序。然后

GLMC_l348.tbl ReadMe test glmc 在作为参数被执行。

最后，step1 中，还会用 Vi，调用 script1_end 脚本对修改过的参数文件 step1_file_putdata 进行处理，这一步就是将 script1 脚本添加的./putdata1 内容去掉。保持 step1_file_putdata 的原状。也就是说用户原来的输入文件不变。

此命令程序则是我这项工作之前需要用户输入的内容。具体来说，在第一步中，原来用户需要输入的是：

```
java      -cp      /home/gaodan/intodb/      Dbreadme.Putdata
/data0/gaodan/glmc/GLMC_l350.tbl /data0/gaodan/glmc/ReadMe test
java      -cp      /home/gaodan/intodb/      Dbreadme.Putdata
/data0/gaodan/glmc/GLMC_l348.tbl /data0/gaodan/glmc/ReadMe test
```

其他各个步骤的设计都与此类似，首先包装原有的命令行产生对应的以待接收参数的文件（即功能文件）。然后构造对应的 Vi 脚本文件，用以处理用户提供的参数文件。最后构造 step 文件，统一调用这些脚本。

五个步骤各自的功能如下：

1. 自动入库

根据星表 ReadMe 文件，自动建表、入库、将时分秒形式转换成度（RAdeg、DEdeg）、建索引、建 id_htm 主键。

2. 自动建 HTM 索引

根据数据表的坐标数据（度，J2000）计算出对应 HTM 索引的 pcode 值，为星表建立 HTM 索引,将 id_htm 主键和 pcode 值两列新建 HTM 索引表。

3. 交叉证认

根据两个数据表及其 HTM 索引表、误差半径进行交叉证认，得到交叉证认表，并分别给交叉证认表中的两表 id 列建索引，方便进一步的数据参数提取。我们为不同的交叉证认需求提供了不同的程序。生成的交叉证认表中只有交叉证认两表的赤经、赤纬、id_htm 主键以及角距离 d 的信息。

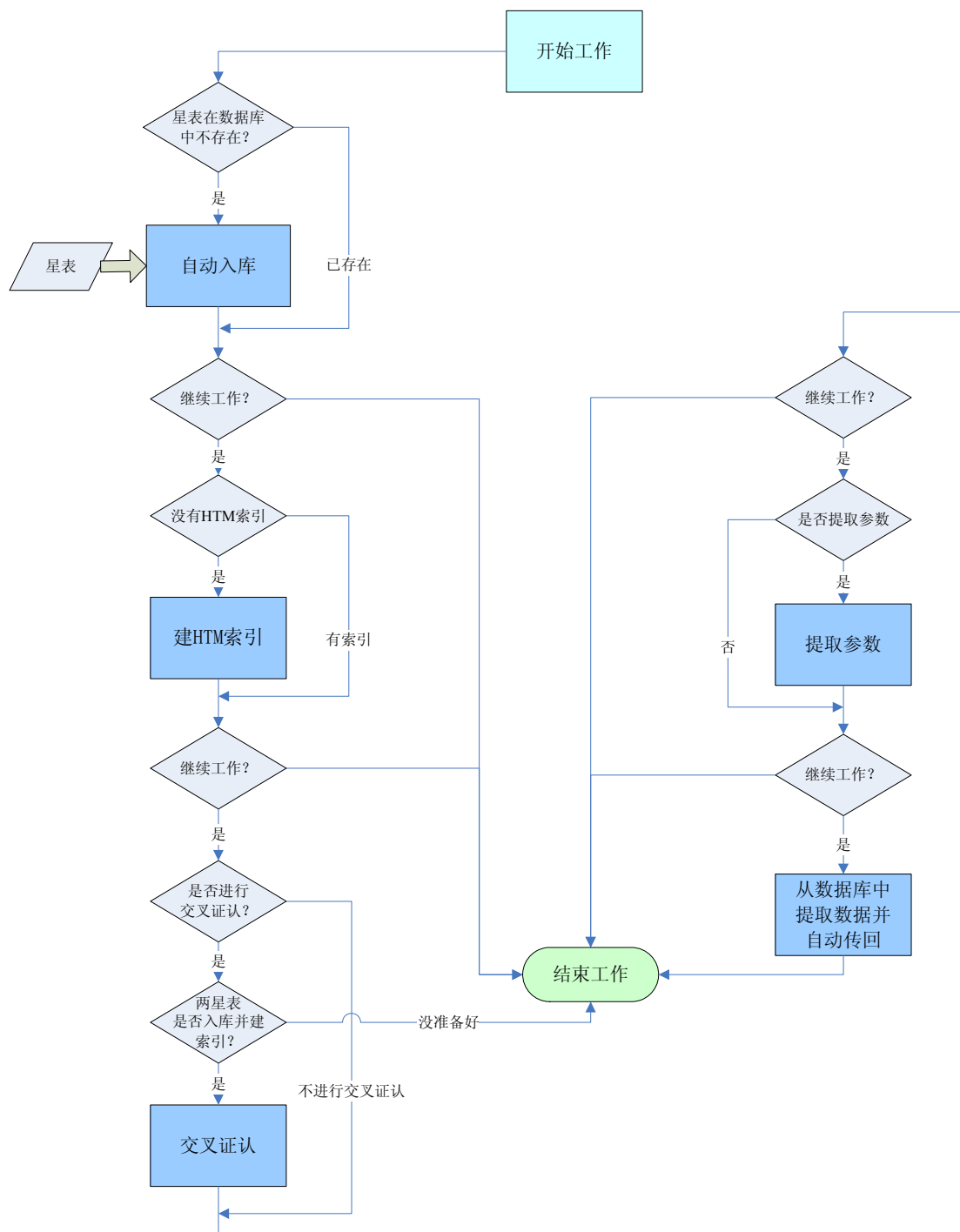
4. 交叉证认参数提取

根据用户需要的参数，对数据库进行操作，生成交叉认证提取参数表。表中包括用户需要的所有参数信息。

5. 数据提取并返回

指定数据库名、表名等信息，自动从数据库中取出数据，并自动通过 FTP 传到用户自己的电脑上。

五个步骤的流程逻辑如图表 24所示。



图表24 海量星表数据处理工具的流程图

5.3 FITS 文件入库系统的设计

5.3.1 意义

FITS (Flexible Image Transport System)是天文学界常用的数据格式，它专门

为在不同平台之间交换数据而设计。1982 年，国际天文联合会 (IAU) 确定 FITS 为各天文台之间用于数据传输、交换的统一标准格式。

FITS 描述了数据定义和数据编码的一般方法，被用于多维矩阵数据（比如一维光谱、二维图像、三维数据立方体）、表列数据的传输、分析和存档。它是与机器无关的，提供了图像的单值转换，精度包括符号在内可以达到 32 位。

当 FITS 中的数据是星表的时候，要将此 FITS 文件包含的星表数据进行处理，常常需要首先将此数据灌入数据库中。因为我们已经开发了 ASCII 文件自动入库的工具。所以我们只需要将 FITS 文件中的星表数据提取、转换为 ASCII 星表文件即可。举例而言，本章上一部分，海量星表数据处理工具就需要此功能进行 FITS 星表的预处理工作。

另一方面，当我们只关心此星表的某些列，该工具能够按照用户指定列，只提取这些列，然后连接 ASCII 自动入库的程序，即可只将用户感兴趣的数据灌入数据库。

必须指出的是，此工具的特点还在于处理非常大的 FITS 文件，特别是将大于 500M 的 FITS 文件入库。因为对于体积庞大的 FITS 星表文件，其星表的行数往往在 10 万量级甚至百万量级。使用现有的办法，就会把特定列全部读取到内存，会导致内存不足的问题。

5.3.2 设计与实现

入库大型 FITS 文件的思路就是将大型文件切割成多个小文件，然后分别读取，再将读取后的内容加以合并。这里的切割是按照行来进行切割。

切割功能由此工具调用 `fitscopy` 来实现，`fitscopy` 是由 CFITSIO 提供的一个 linux 下的命令程序。CFITSIO[31] 的是 C 语言的 FITS 文件输入输出 (IO) 工具集。

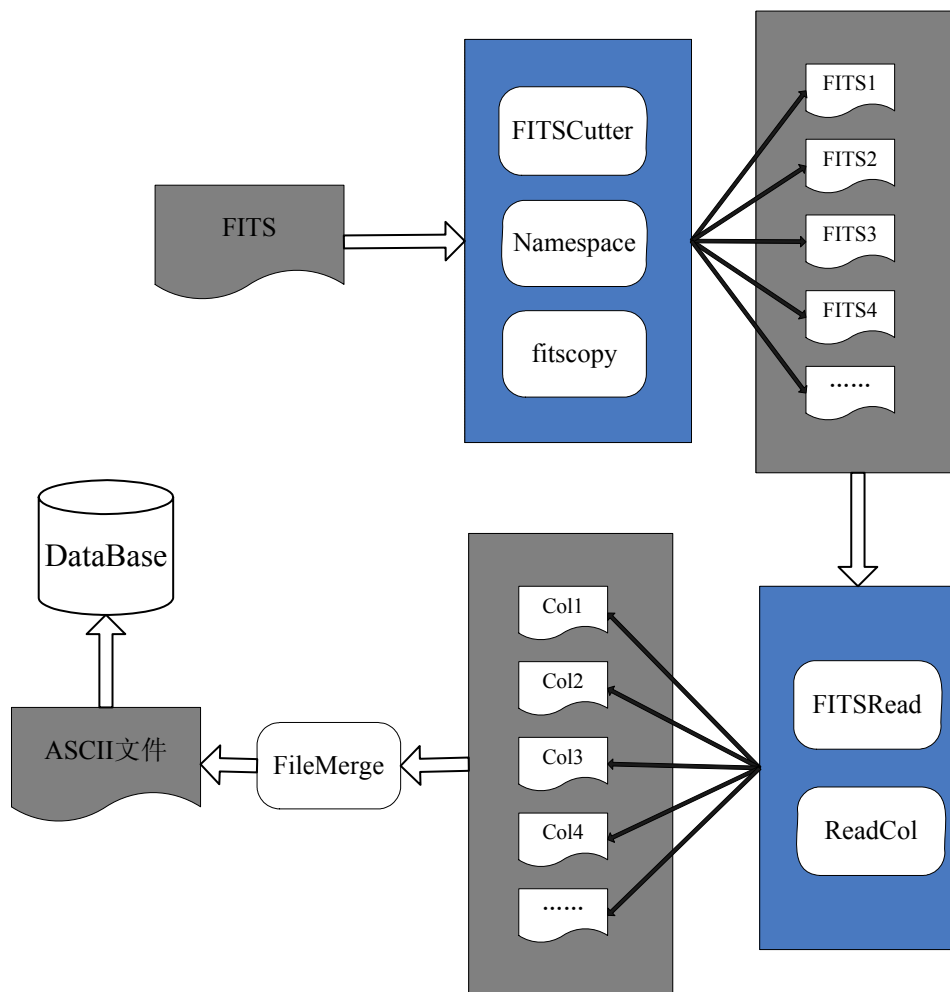
我提供的工具首先根据 FITS 文件的行数，自动判断要将原 FITS 文件切割成多少个小文件，并对这些小文件的文件名进行控制。然后根据用户提供的参数，在各个小文件抽取用户需要的列，每列分别保存为一个临时文件。最后，将这些列文件逐个逐行读取，合并为一个文件，就是用户需要的 ASCII 格式的 FITS 星

表文件，调用本章上一节提到的海量星表处理程序，即可将此文件入库。

本系统被分成以下模块，Namespace、FITSCutter、FITSRead、ReadCol、FileMerge 等。其中 Namespace 主要是对文件名进行控制，FITSCutter 根据文件大小，生成调用 fitscopy 进行切割的批处理文件，FITSRead 和 ReadCol 进行 FITS 列的抽取，FileMerge 则将小文件进行合并。

5.3.3 流程

本系统在linux上运行。程序运行流程如图表 25所示。



图表25 FITS 文件入库工具流程图

6 总结和展望

本文是对于虚拟天文台数据研究方面的一篇论文。中心是虚拟天文台数据访问服务的注册系统，除此之外，还介绍了本人所做的其它相关工作。

本文第一章比较具体地介绍了虚拟天文台的概念、现状以及虚拟天文台数据访问服务（VO-DAS）的研究现状和意义。

第二章着眼于本系统涉及到的天文数据标准。按重要性介绍了可扩展标识语言（XML）、VOTable 和天文数据查询语言（ADQL）

第三章介绍的是虚拟天文台注册系统。首先研究了注册系统的基石--国际虚拟天文台联盟（IVOA）制订的相关标准。然后对几个较为成功的虚拟天文台注册系统（VO Registry）进行了介绍。

第四章研究了虚拟天文台数据访问服务（VO-DAS）的注册系统。从此系统的功能、涉及实现以及与 VO Registry 的互联方面进行了详细的阐述。

第五章则介绍了本人在读研期间进行的三个与虚拟天文台数据有关的工作。它们分别是：与 VO-DAS 相关的图像和光谱的联合查询；用于处理星表的海量星表数据处理工具以及用于星表格式转换的 FITS 文件入库系统。

目前，国际上关于虚拟天文台的各项研究方兴未艾、蓬勃发展。中国虚拟天文台的数据访问服务也刚刚完成试验运行。此服务现在正在进行较多的完善和功能扩展，此服务的开发必将是个逐渐完善的过程。而此服务的注册系统也必将随着此服务的改进而进行进一步的调优和完善。同时，IVOA 关于资源元数据和注册接口的相关标准现在也没有进入定稿阶段，可能还会出现一些改动，VO-DAS 注册系统也会进行相应的跟进，以保持与国际标准的一致。

另外，本文介绍的光谱与图像的联合查询方面，仅仅进行了初步设计，还有

待进一步的研究和实现；而海量星表数据处理工具已经测试运行了数月，功能方面较令人满意，而用户界面、容错性等方面值得改进的地方仍很多；FITS 文件入库系统现在实现了星表格式的转换，也能够使用海量星表数据处理工作进行入库等进一步工作，但是自动化程度不高，值得进一步研究、改进。

参 考 文 献

- [1] Szalay A, Gray J. The World-Wide Telescope. Science, 2001, 293: 203
- [2] 赵永恒. 互联网时代的天文学革命: 虚拟天文台. 科学, 2002, 54(2): 13
- [3] 美国国家虚拟天文台, <http://www.us-vo.org/>
- [4] IVOA, <http://www.ivoa.org/>
- [5] 崔辰州, 虚拟天文台系统设计[博士学位论文], 中国科学院研究生院, 2003
- [6] LAMOST, <http://www.lamost.org/>
- [7] Globus Alliance, <http://www.globus.org>
- [8] D.Gao, Y.Zhang, and Y.Zhao, A system integrated with query, cross-matching and visualization, Proceedings of the SPIE, 2006, 6274:627414
- [9] 万维网联盟, <http://www.w3.org/>
- [10] XML, <http://www.xml.org/>
- [11] VOTable, <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOTable>
- [12] ADQL, <http://www.ivoa.net/Documents/latest/ADQL.html>
- [13] SAX, <http://www.saxproject.org/>
- [14] DOM, <http://www.w3.org/DOM/>
- [15] FITS, <http://fits.gsfc.nasa.gov/>
- [16] Resource Metadata, <http://www.ivoa.net/Documents/latest/RM.html>
- [17] Registry, <http://www.ivoa.net/Documents/latest/RegistryInterface.html>
- [18] 元数据, <http://www.w3.org/Metadata/>
- [19] Weasl, <http://weasl.sourceforge.net/about.html>
- [20] Carnivore, <http://nvo.caltech.edu:8080/carnivore/>
- [21] Java, <http://java.sun.com/>
- [22] XQuery, <http://www.w3.org/TR/xquery/>
- [23] NASA, <http://www.nasa.gov/>
- [24] Skyview, <http://skyview.gsfc.nasa.gov/>
- [25] 北京天文数据中心, <http://badc.lamost.org/>
- [26] 斯特拉斯堡天文数据中心, <http://cdsweb.u-strasbg.fr/>
- [27] 高丹, 张彦霞, 赵永恒. 天文星表入库的自动化. 天文研究与技术, 2007, 4(1):53-58
- [28] SDSS, <http://www.sdss.org/>
- [29] 2MASS, <http://pegasus.phast.umass.edu/>
- [30] Vi, <http://www.vim.org/>
- [31] CFITSIO, <http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html>

发表文章

- 路勇, 刘超, 崔辰州, 赵永恒. VO-DAS Registry 系统的设计与实现, 天文研究与技术, 2007 (已接收)
- 高丹, 路勇, 张彦霞, 赵永恒. 海量星表融合系统 (XMaS_VO) 的设计与开发, 天文学进展 (已投稿)
- Chao Liu, Hai-Jun Tian, Dan Gao, Yang Yang, Yong Lu, et al. VO-DAS: Integrated Access of Distributed and Heterogeneous Astronomical Data Resources. (撰写中)

致 谢

三年的硕士学习生活即将结束,在此我要向给过我指导、帮助和关心的老师、同学和亲友表示我衷心的感谢。

感谢我的导师罗阿理副研究员和赵永恒研究员。感谢罗老师,三年前的夏天,罗老师是我来到国家天文台认识的第一位老师,他让我对 LAMOST 项目有了最初的认识。特别感谢他一直以来对我工作和生活上的关心和照顾,感谢他给我在论文工作上提供的帮助。

感谢 LAMOST 项目的总经理赵永恒研究员。赵老师富有人格魅力和科研洞察力,知识渊博,思维敏锐,对待科研一丝不苟,对待学生更是平易近人。感谢赵老师对我的指点和关心。赵老师的开阔的胸襟和为人师表的风范是我做人的榜样。

感谢虚拟天文台项目组的崔辰州老师。感谢他将我引入虚拟天文台的大门,让我领略到天文学,特别是虚拟天文台的迷人魅力。感谢他在本论文从选题到最后完成过程中对我的指导和帮助。他给我的课题指明方向,并悉心指导,提供了优良的软硬件环境和学习机会,十分感谢崔老师对我分析问题、解决问题能力的培养。

感谢 LAMOST 项目组的张彦霞、张昊彤、王刚、陈建军、石火明、陈英、袁晖等老师的帮助。尤其感谢张彦霞老师对我工作和生活的很多照顾,以及在我入党的过程中,给予我的思想上的指导。

感谢国家天文台西部选址项目组的姚永强首席研究员、王俊杰老师、杨国安老师、房礼刚师傅等,感谢他们给我的所有关心和帮助。尤其感谢姚永强老师对我在科研方面的引导和在生活方面的帮助。还要感谢我一起在西部工作过的马江龙、石俊、施洋等。西部生活永难忘。

感谢国家天文台研究生管理办公室杜红荣老师、艾华老师。感谢她们在我读研期间对我研究生生活和工作上的帮助，她们的无私工作为天文台的研究生们创造了优越的生活和学习环境。在此向她表示衷心的感谢。

感谢党支部孙盛慈老师、李金增老师和张彦霞老师，孙老师慈祥可亲，李金增老师为人正直，张彦霞老师平易近人。他们引领我在入党的道路中克服重重困难，以致最后修成正果，成为一名光荣的共产党员。衷心感谢他们。

感谢 LAMOST 项目组的高丹同学，她对我在科研思路方面以及具体工作上都给予了我非常多的帮助。她和我一起完成了本文提到的海量星表数据系统的很多工作，她在 VO-DAS 项目中，也同样给予我非常多具体的帮助。谢谢她。

感谢虚拟天文台项目组的刘超博士，他在项目管理、设计以及具体的程序实现上富有经验。在 VO-DAS 项目中，他担当了重要的角色，对我的工作帮助很大。谢谢他。

感谢英国虚拟天文台 AstroGrid 项目组的秦玲女士，感谢她在我在进行 VO-DAS Registry 与 AstroGrid Registry 相连接的过程中给我的无私而热情的帮助。

感谢 LAMOST 项目组与我一同学习和生活的同学们。有他们的陪伴，我的学习和生活多了很多乐趣。他们是：高丹、杨阳、罗宇、刘超、田海俊、李会贤、亓大志、孙华平、尹红星、郑征、胡晨、吴悦、李丽丽、姚嵩、薛元、朱黎楠、孙士卫等。特别感谢田海俊、罗宇和亓大志对我工作的帮助。

感谢国家天文台曾经关心、帮助、给我提出过建议的所有同学。特别是刘国卿、李晓昕、朱轶楠、刘健、黄静、孔令军、范舟、郑位康、裴少芳等同学。有了他们，我这三年的生活更有趣而精彩。

感谢中国科学院研究生院心理协会的所有朋友们，这个协会是我研究生生活中最重要的部分之一。和他们在一起，让我充满乐趣和自信。感谢心理学，它让我距离永难触及的终极真理，近了一步。

感谢我的家人和朋友。最真挚地感谢我的父母赋予我以生命，抚育我长大，教育我做人，感谢他们给我的一切却不求回报。感谢我的姐姐一直以来对我的包

容和照顾。最深地感谢我已过世的最疼爱我的曾外祖母,看着我走过淘气的童年、看着我走进叛逆的少年,看着我走出家乡,却没有等到我回来。