

分类号 _____
UDC _____

密级: _____
编号: _____

中国科学院研究生院

硕士学位论文

中国虚拟天文台可视化设计及实现

邵惠娟

指导老师: 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: 天体物理

论文提交日期: 2004 年 6 月 论文答辩日期: 2004 年 7 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: 阎保平

中国科学院研究生院
研究生论文

中国虚拟天文台可视化设计及实现



作 者	邵 惠 娟
指导老师	赵 永 恒
学科专业	天 体 物 理
研究方向	天 文 数 据 处 理 方 法
申请学位	硕 士
培养单位	国 家 天 文 台

二〇〇四年六月

**National Astronomical Observatory
Chinese Academy of Sciences**

**Design and Realization of
Visualization for
Chinese Virtual Observatory**



Huijuan Shao

Advisor: Prof. Yongheng Zhao

June 2004

摘要

随着地面和空间观测站的建立、探测器灵敏度的提高、望远镜口径的增大，天文界正面临着“数据雪崩”的问题。如何通过现代计算机软硬件和网络系统有效地对这些 PB 量级的数据进行存储、计算、统计、分析和显示，成为天文学面临的一大问题。在这样的背景下，虚拟天文台应运而生。各国虚拟天文台正致力于通过数据网格来实现海量分布式数据的管理和统一访问等功能。

数据可视化是虚拟天文台中必不可少的一部分。它能够从大量的数据中提取有用的信息，或者得到其他方式的不容易察觉的数据特征，以用户需要的形式展现给用户。最主要的是它能够通过网络，通过表格、图形、图像、视频甚至虚拟现实方式给用户以直观的感受。

本文着重阐述了对天文星表数据和天文图像的可视化设计及实现方法。天文星表包括 Hipparcos, Tycho2, GSC1.2, GSC2.2, USNO A2.0, USNO B1.0, UCAC2 等天文学家常用的星表；星图包括 DSS-I 全天星图。

在设计过程中，本文首先对可视化表和图进行了初步调研，并进行了需求分析和系统设计；再根据虚拟天文台提供的数据服务结合现有的计算机应用技术分析、比较、选择，提出了实现可视化功能的解决方案。

可视化服务实现的基本功能主要包括：利用基于流方式的 SAVOT 工具对 VOTable 格式数据进行解析；应用支持任何格式的 JAI 框架来实现对 FITS 图像的支持；用 AWT、Swing、JAI 包实现图像显示和处理的功能；用 Apache、Tomcat 提供 Web 服务功能；用 Java 提供的 net 包实现了 Servlet 和 Applet 之间的通信；基于网格环境 Globus Toolkit 工具以通知框架初步实现了数据服务。

可视化工具侧重于开发和应用，星表和星图的可视化仍需要优化和完善。除此之外，虚拟天文台还需要进一步实现对天文光谱、交叉证认等的可视化，来充实和发展对科学研究和普通用户有用的可视化工具箱。

关键字：可视化 虚拟天文台 VOTable Applet

Abstract

Along with the establishment of the ground-based and space-based observation stations, the enhancement of the sensitivity of the detector, as well as the increase of the telescope caliber, the astronomy community is confronted with the data avalanche problem. The community is facing how to efficiently store, calculate, count and display of the data on the PB scale through the modern computer hardware and software and network systems. Based on this status, the Virtual Observatory (VO) emerges as the times require. The virtual observatory of the world is working on realizing the management and access interface of the vast distributed data through the data grid.

Visualization is the absolutely necessary part of the virtual observatory. It can get useful information from mass data, or get the data feature which can't be observed by other ways, and display to the users by the required form. The most important thing is that it can give the users the direct feeling by the sheet, graphics, image, video, and even virtual reality.

In this thesis we are mainly discussing the visualization design and implement method of astronomical star tables and images. The star tables include the Hipparcos, Tycho2, GSC1.2, GSC2.2, USNO A2.0, USNO B1.0, UCAC2 and others generally used by astronomers. The images include the all-sky star images DSS-I.

In this thesis, firstly, I research the visualized sheet and image, analyze the demand, and design the system. Furthermore, I give out the solution for the implement of the virtualization function, according to the analysis, compare and selection of the present computer application technology combined with the data service of virtual observatory.

The basic function of the visualization service includes: parsing the VOTable data by the SAVOT tool based on the stream method; supporting the FITS image by the JAI frameworks which can support all formats; realizing the image handle and display function by the JAI, AWT, Swing packages; affording the web service function by Apache, Tomcat; realizing the communication between Servlet and Applet by the net package from Java; realizing the data service on the notification frameworks by the Globus Toolkit backed on the grid environment.

The thesis emphasizes on the development and application of visualization tools for the star tables and images, which still has to be optimized. Besides this, the virtual astronomical observatory still has to realize the visualization of fiber spectrum, cross-match, which can enrich and develop the useful visualization tool kits for scientific research and general users.

Key Words: Visualization, Virtual Observatory, VOTable, Applet

目 录

第 1 章.	虚拟天文台和天文数据可视化研究现状	1
1.1.	虚拟天文台的产生和背景	1
1.2.	数据可视化的背景和重要性	1
1.2.1.	数据可视化的必要性	1
1.2.2.	可视化的分类及应用	2
1.2.3.	虚拟天文台中可视化重要性	4
1.3.	虚拟天文台可视化研究现状	4
1.3.1.	天文数据可视化现状	4
1.3.2.	虚拟天文台可视化工具的基本要求	5
1.3.3.	中国虚拟天文台解决的问题	6
第 2 章.	可视化系统概述	8
2.1.	虚拟天文台可视化目标	8
2.2.	系统功能说明	8
2.2.1.	对不同数据库的查询	8
2.2.2.	对FITS图的支持	8
2.2.3.	对图像的基本操作	8
2.2.4.	对图形的支持	9
2.2.5.	对多副图像的重叠分析	9
第 3 章.	可视化系统设计	10
3.1.	系统硬件架构	10
3.1.1.	网络结构图	10
3.1.2.	网络结构硬件说明	10
3.1.3.	网络结构图功能说明	11
3.2.	系统设计流程	12
3.2.1.	数据顺序图	12
3.2.2.	数据顺序图说明	12
3.3.	数据服务—可视化数据来源	13
3.3.1.	自动注册	13
3.3.2.	通知框架机制	13
3.4.	星表数据展现 (XML数据到图形显示)	13
3.5.	星图及表的数据展现功能	14
3.5.1.	坐标系统功能说明	14
3.5.2.	FITS I/O	15
3.5.3.	JAI FITS支持	15
3.5.4.	支持其他图像格式	16
3.5.5.	图像处理操作	16
3.5.6.	图像显示小窗口	16
3.5.7.	图像Zoom小窗口	16
3.5.8.	图像面板小窗口	17
3.5.9.	图像图标	17
3.5.10.	图像控制面板	17

3.5.11.	主要图像显示小窗口	18
3.5.12.	选择星体小窗口	18
3.5.13.	HDU显示小窗口	18
3.5.14.	图像颜色管理器	18
3.5.15.	图像处理工具	19
3.5.16.	图像剪切和像素表	19
3.5.17.	图像作图	19
3.5.18.	图像作图编辑	19
3.6.	客户端和服务端功能	19
3.6.1.	程序存放端的选择	19
3.6.2.	服务器端处理内容	20
3.6.3.	客户端功能	20
第4章.	可视化服务的实现及关键技术	22
4.1.	系统说明	22
4.1.1.	开发工具的选择	22
4.1.2.	关键技术的说明	23
4.2.	软件架构	24
4.2.1.	数据流图	24
4.3.	GLOBUS TOOLKIT的应用实现	25
4.3.1.	网格服务接口描述	25
4.3.2.	实现的基本步骤及部署自动注册	26
4.3.3.	Globus Toolkit3 提供的两种实现方法	26
4.3.4.	日志	28
4.3.5.	服务数据(ServiceData)	28
4.3.6.	通知框架机制	29
4.3.7.	数据服务的初步实现	31
4.4.	WEB服务器端的APACHE与TOMCAT	35
4.4.1.	Web应用程序目录结构及错误处理	35
4.4.2.	Apache与Tomcat连接	36
4.4.3.	用户session及图像转换	37
4.5.	客户端与服务端(APPLET与SERVLET)通信方式对比选择	37
4.5.1.	通过HTML页面传递Applet参数	38
4.5.2.	用java.net包建立双向通讯	38
4.5.3.	远程方法调用RMI	39
4.5.4.	选择和实现	39
4.6.	VOTABLE与XML的JAVA解析	41
4.6.1.	FITS说明及显示说明	41
4.6.2.	VOTable格式数据	42
4.6.3.	支持 Document Object Model (DOM) 的解析器	43
4.6.4.	支持 Simple API for XML (SAX) 的解析器	43
4.6.5.	VOTable解析在Web服务器端的实现	44
4.7.	数据图形	46
4.7.1.	世界坐标系及实现	46
4.7.2.	FITS对JAI支持实现	46

4.7.3.	用户界面	47
4. 8.	CONESEARCH例子	48
第 5 章.	总结和展望	52
致 谢	53	

图 表 目 录

图 1-1 可视化模型	2
图 3-1 虚拟天文台网络结构图	10
图 3-2 数据顺序图	12
图 3-3 JAI图像处理和显示	16
图 4-1 各部分关键技术说明	23
图 4-2 可视化数据流图	24
图 4-3 一个网格服务所需要的全部包和文件	26
图 4-4 通过继承的网格服务实现	27
图 4-5 通过代理的网格服务实现	27
图 4-6 服务数据 (SERVICE DATA) 在OGSA中的结构	29
图 4-7 OBSERVER/OBSERVABLE PULL模式	30
图 4-8 GT3 通知订阅	31
图 4-9 接口流程	32
图 4-10 数据服务客户端与服务器端交互顺序图	33
图 4-11 网格服务客户端程序模型	33
图 4-12 网格服务服务端程序模型	34
图 4-13 用户SESSION处理	37
图 4-14 APPLET访问SERVLET数据流程	40
图 4-15 INPUTPANEL类图	41
图 4-16 类INPUTRESTRICT	41
图 4-17 SAX基于回调的解析机制	44
图 4-18 VOTABLE解析处理	45
图 4-19 用户输入坐标和世界坐标系	46
图 4-20 从源图像到显示图像的处理	47
图 4-21 用户界面功能类图	48
图 4-22 CONESEARCH主界面图	49
图 4-23 CONESEARCH数据表图	50
图 4-24 CONESEARCH中的VOTABLE数据	50
表格 3-1 星图及表的数据展现组件要求	18
表格 4-1 VOTABLE解析选择	45
表格 4-2 JAI对FITS支持要求的变换点	47

第1章. 虚拟天文台和天文数据可视化研究现状

1.1. 虚拟天文台的产生和背景

200 多年来,天文学研究的方式一般是分别由单个天文学家或几个天文小组对数量相对较少的天体进行研究。近十来年,各种技术正在经历史无前例的飞速发展,望远镜口径的增大、大面积探测阵的发展,天文学正在步入覆盖较大波长范围、大天区数字巡天的新时代。这些巡天数据通常以TB量级来进行计算^[1]。

这些天文数据需要研究人员对天体进行编目整理工作,例如 CDS 和其他数据中心的研究人员开发出各种工具来对他们的数据库进行分析。

但是各个巡天项目之间的数据整理是不同的,而且每个地方的天文科学家在进行研究时都需要做所需数据的下载,大的数据量还需要镜像,这给需要进行研究的天文学家的硬件提出了比较高的要求,而且还造成了硬件资源的重复和浪费。欧洲南方天文台(ESO)数据管理部门负责人 Peter Quinn 表示,虚拟天文台的主要职能是将十多个巡天项目产生的海量数据融合到一起。

针对上述的情况,各国纷纷进行虚拟天文台的研究,以解决数据雪崩带来的问题。目前正在进行的虚拟天文台项目有:美国的国家虚拟天文台(NVO)、英国的AstroGrid(UKVO)、欧洲天体物理虚拟天文台(AVO)、加拿大虚拟天文台(CVO)、印度虚拟天文台(VO-India)、德国天体物理虚拟天文台(GAVO)、澳大利亚虚拟天文台(Aus-VO)、日本虚拟天文台(JVO)、俄罗斯虚拟天文台(RVO)、韩国虚拟天文台(KVO)^[2]等。

这些项目将采用一个单一、易用的接口把现有的数据融合起来。很多情况下,在虚拟天文台把数据返回给研究人员之前,它都不清楚虚拟天文台在检索哪些数据库。虚拟天文台将通过单一的访问点实现对大量数据库的访问,这将给那些小机构中的研究人员提供更多的机会。另外,把不同的数据综合起来也将增大天文研究的可能性。Djorgovski认为,“这些空前数量和质的数据会促使你提出创新性的问题,褐矮型搜寻就是一个例子”^[3]。

1.2. 数据可视化的背景和重要性

1.2.1. 数据可视化的必要性

有关研究数据表明,在味觉、听觉、触觉和视觉等多种信息感知形式中,通过视觉获取的信息量占到总量的 60%以上。人类对可视化的研究和探索从远古以来就已存在,诸如“百

闻不如一见”的成语深刻地体现了人类对可视化重要性的认知程度。

可视化改变了数据的“面貌”，强化了数据的可读性，使之更为形象直观，图文并茂。首先它应用依赖人的感知系统天生就能迅速理解的基本特征：颜色、大小、形状、运动和邻近度。

其次，可视化工具扩展了人的视觉功能，海量的数据通过可视化变成形象，现出隐藏的规律，能够激发人的形象思维。设计人员对大量数据进行特征分析，并将它可视化展现出来，使用这些工具的人们就能够找到、分析数据内部隐含的相关联系，看出表明趋势的数据模式、识别数据差异、发现数据的异常值或错误、准确找到最小值和最大值以及识别聚类，或者发掘以往不可能知道的信息。

另外，可视化已经不仅仅局限于对已有的数据或者事物的真实“再现”，还可以将原本不“完全”存在甚至“不存在的事物”制造“出来，以模拟、仿真其可能的表现和发展趋势，并进一步分析数据中隐含的规律，为科学决策提供依据^[4]。

再次，可视化能够帮助人们增强认知能力。Xerox Palo Alto 研究中心学者 Card 等人提出了可视化可以增强认知的六种主要方式：通过扩大用户可用的存储量和可处理的资源量；通过减少对信息的搜索；通过使用可视化表示提高对模式的发掘；通过调动感知推理操作；通过把感知注意用于监控；通过把信息编码为可操作的媒介。

在现代科学的各个学科中，人们要从大量的数据中提取有用的信息、了解数据之间的相互关系和发展趋势，可以求助于可视化技术。可视化是发现和理解科学计算过程中各种现象的有力工具，它可以大大加快数据的处理速度，使目前不断产生的数据得到有效的处理；它可以在人与数据、人与人之间实现图像通信，而不是目前的文字通信或数据通信；它还可以使人们对计算过程实现引导和控制，通过交互手段改变计算依据的条件观察其影响^[5]。可视化参考模型如图 1-1^[5]。

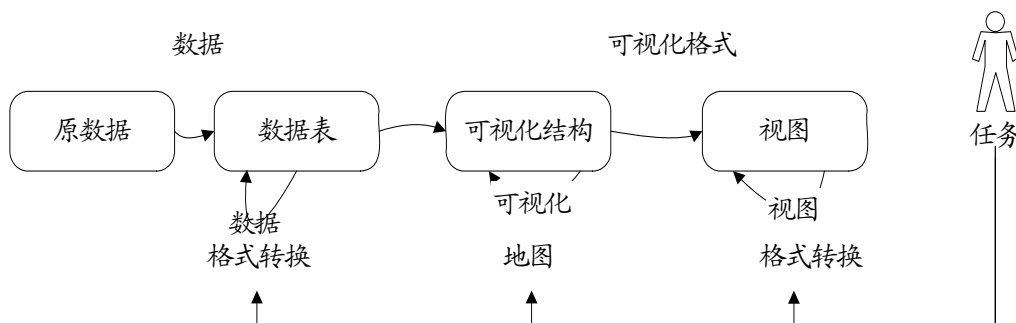


图 1-1 可视化模型

针对不同的学科，设计和选择什么样的显示方式，使用户深刻地了解庞大的多维数据和它们相互之间的关系，是数据可视化的重点和核心。

1.2.2. 可视化的分类及应用

可视化应用有几种方法，但最基本的方法是根据可视化应用的基本数据类型进行分类。

由本·施奈德曼(Ben Shneiderman)概述的按数据类型进行归类的方法把数据分成以下七类：一维数据、二维数据、三维数据、多维数据、时态数据、层次数据和网络数据。

1. 一维信息可视化

一维信息是简单的线性信息，如文本，或者一系列数字。

2. 二维信息可视化

在信息可视化环境中，二维信息是指包括两个主要属性的信息。例如天文中，经度和纬度两个属性值就能够描述天体在天图中的位置。

3. 三维信息可视化

三维信息引入了体积的概念，它能够表示现实的三维物体。在现实生活中，我们不可能像看到周围的生活空间一样看到太空中天体的位置、以及某个天体活动对周围天体或物质的影响，通过三维模型可视化可以模拟出这些效果来。

4. 多维信息可视化

多维信息是指在信息可视化环境中的那些具有超过三个属性的信息，在可视化中，这些属性的重要性是相当重要的。天文数据的属性值是很多的，每个重要的属性都可以作为多维信息中的一维，通过其中几项属性不同变化模拟出对整体或局部结果的影响，直接或间接得出一些科学结论。

5. 时态数据的可视化

以图形的方式显示随着时间不断产生的数据是可视化信息最常见、最有用的方法之一。

6. 层次数据的可视化

层次数据即数形数据就是有这样一种内在结构的数据：每个项目或节点都有一个父节点(最上面的节点即根结点除外)。节点分兄弟节点(拥有同一个父节点的节点)和子节点(丛属某个父节点的节点)。层次结构相当常见。商业组织组织、计算机数据存储系统和家谱图都是按数形结构排列的层次数据。

对大型层次结构而言，Windows Explorer 及其它应用采用的树形视图格式无法用一个视图表示整个结构。如果子节点缩在父节点里面，一个视图就能显示整个结构。

层次数据的三维视图也会遇到这样的问题：每个节点的大小和内容都是隐藏的。尽管你可能知道某个节点在结构中的位置，但所有节点似乎都一样。利用树形图(treemap)来显示层次数据可以解决这个问题。树形图用嵌套矩形来表示数据层次。所有矩形都放在一个很大的边界矩形里面，里面的每个矩形表示一个节点。如果是父节点，里面还会有子节点。这种设计使树形图用一个视图就能够显示数据层次里面的所有节点。实际上，树形图显示的节点数量要比传统数形视图多出一个量级。

除了一个视图就能表示层次里面的所有节点，树形图还能在同一视图显示单个节点的信

息。树形图中矩形的大小表示了它在整个层次中的相对大小，其它属性由颜色和内容敏感的属性显示区加以表示。譬如说，用树形图来表示按杜威图书分类法排列的一堆书籍。矩形的排列表示层次中该级别的书籍的数量，而矩形里面的颜色表示该级别书籍的翻阅频率。

7. 网络数据可视化

网络数据指与任意数量的其他节点有着关系的节点。网络数据集里面的节点不像层次节点，层次节点都有惟一的父节点，而网络数据因为属性和节点之间的关系可能非常复杂，如果不用某种可视化方法，网络数据很难显示。比如说，因特网上面有成千上万台服务器，服务器之间又可能存在众多路径。尽管对特定的任务而言，通过观察表格和统计数字，有可能了解网络流量模式，使用量高峰和低峰以及节点之间的备用路径，但使用视觉表示法却可以大大简化这项复杂的工作。

1.2.3. 虚拟天文台中可视化的重要性

在虚拟天文台系统中，用户和计算机之间的界面是必不可少的，它能够从大量的数据中提取有用的信息，以用户需要的形式返回给用户，或者得到其它方式不容易察觉的数据特征，或者将数据定量分析的结果以直观的方式反映出来，或者按用户自己的意愿增减部分属性，为用户对数据的全面的理解提供方便^[6]。一幅画胜过千言万语，可视化的直观性即说明了计算机和用户之间这座桥梁的重要之处。

更加重要的是，虚拟天文台上基于互联网的 Web 图形化的浏览界面让我们从更多的角度、更深入地看到了世界的另一面--丰富而复杂的信息资源。人们发现，与传统静态的文本、数字相比，有时用图形、图表、视频甚至虚拟的现实场景等可视化形式表现信息更加高效。

虚拟天文台框架允许软件提供者提供兼容的软件工具或用户接口，用户要应用这些工具无需研究虚拟天文台后台系统，只需要根据这些工具的要求写客户端接口程序即可。可视化工具是在虚拟天文台框架上提供的有标准接口的软件之一，它能够提供针对不同类型和格式的数据、实现不同的可视化功能。可视化是虚拟天文台各种服务中的一种。

1.3. 虚拟天文台可视化研究现状

1.3.1. 天文数据可视化现状

天文数据包括的范围很广，按波段分为射电、红外、可见光、紫外、X 射线、Gamma 射线等，按数据功能类型有光谱、星表、星图等。可视化要针对不同类型的数据，分别进行分析和展现。

1. 对星表数据进行处理。根据不同的应用，天文观测得出的星表数据的可视化，可以做简单处理，也可以做得比较复杂和完善。平常我们在互连网上查到的数据，就是经过一定的

处理得到的。这样的可视化比较简单，它根据星表的数据属性，将从数据库或文件中查到的数据用 HTML 或者一些脚本语言如 JSP、ASP、PHP 编写，二维地放在 Web 页面中。

此外，我们可以根据星表数据的属性值进行三维数据可视化，用来显示在天空中的三维坐标，做虚拟现实场景；或者对其中重要的属性进行多维显示，以找出其中的特征，从而发现有用的天体等等。

2. 对星图数据进行处理。天文上的观测图像主要是 FITS 图。

首先要对这些 FITS 进行处理后展现给用户看，即二维可视化，以及与其他通用图像格式如 JPEG, GIF, PNG 之间提供转换。

另外，对这些天图，可视化可以做一般的图像处理，比如说放大、缩小、颜色上的处理。

3. 对光谱数据进行处理。

4. 数据挖掘的可视化。

在虚拟天文台可视化研究初期，我们将主要针对基于已有的下列星表和星图数据进行可视化。

星表: Hipparcos, Tycho2, GSC1.2, GSC2.2, USNO B1.0, USNO A2.0, UCAC2

星图: DSS-I 全天星图

对于星图和星表的可视化软件，很多观测中心都开发出了自己的软件，以展现各数据中心的的数据。主要的影响力比较大的软件工具有：

CDS用Java^[7]开发的Aladin^[8]。Aladin的主要功能包括能够查询CDS中的大部分星表和星图，最重要的一点是能够将不同波段的星图和相应的星表叠加在一起给用户直观的感受，从而进行天文方面的研究。Aladin有三种不同的版本：网络版的Applet，应用程序和专门用作查询图的简体版。

JSky^[9]，JSky是欧南天文台开发的应用程序软件，它实现了Aladin的大部分功能。这个软件是开发源码的。

印度天文台开发的VOPlot。VOPlot主要针对VOTable^[10]数据格式的二维图形化显示，包括Applet版本和Application版本。

用 Tcl/Tk 开发的 SAO Image DS9。

这些可视化软件工具实现的功能有重叠之处，也有各自不同的特色。可视化工具已经有了一定的基础和发展，我们将基于前人的工作，根据虚拟天文台的具体需求，尽量实现期望的可视化功能。对于已开发出的开源的稳定模块，若有借鉴作用，我们将根据具体情况进行采用或者修改其中的部分内容，以便减少模块、代码的开发工作。

1.3.2. 虚拟天文台可视化工具的基本要求

虚拟天文台允许天文学家和研究人员直观和全面地远程操作由各种巡天观测产生的大量数据，在 TB 量级上对图像、星表、光谱等数据进行分析。但在对这些数据可视化时，只有少

量的数据会传送并给实时使用，主要是因为：

1. 在用户需求上，大量数据例如半个天区的查询，可以直接提供下载，由本地的工具进行分析。对大量数据的实时显示的需求是很少的。

2. 在实现上，对通信而言，大量的数据会产生时间延迟现象，少量的数据利于传输；对计算机而言，大量数据可视化会给用户系统带来很大的负担，特别是内存的开销，少量的数据方便实现。

因而可视化首先要求数据量不大，这样一般的个人计算机系统能够处理，数据的传输也能够承受，用户可以使用。

1.3.3. 中国虚拟天文台解决的问题

中国虚拟天文台可视化的开发的不同之处，在于我们将基于网格平台的开发，可视化将作为其中的一项服务提供给用户，在性能上能够开发出达到用户基本要求的工具。

参 考 文 献

- [1] 张彦霞,赵永恒. 天文学中的数据挖掘和知识发现. 天文学进展, 2002, 20(4): 312
- [2] 崔辰州. 中国虚拟天文台系统设计:[学位论文]. 北京: 中国科学院研究生院
- [3] Geoff Brumfiel. The Heavens at your Fingertips. Nature, 2002(420): 262-264
- [4] 清水. 计算机世界. 信息可视化: 畅游网络空间的伴侣. 2004(11): B3
- [5] 唐泽圣等. 三维数据场可视化. 北京, 清华大学出版社
- [6] [可视化]刘玮. 从认知到感知: 谈信息可视化技术.
<http://media.ccidnet.com/media/ccu/592/05201.htm>, 2003-12-22
- [7] Products and Technologies: Java Technology. <http://java.sun.com>
- [8] [Aladin] The Aladin Interactive Sky Atlas.
<http://aladin.u-strasbg.fr/aladin.gml>
- [9] [JSky] JSky-Java Components for Astronomy.
<http://archive.eso.org/JSky/>
- [10] [VOTable] VOTable Documentation. <http://www.us-vo.org/VOTable/>

第2章. 可视化系统概述

2.1. 虚拟天文台可视化目标

对于可视化工具，从用户的角度来看，要求功能强大、易学易用；从 VO 系统角度看，要求与其它工具兼容性好，与网格环境能够很好的融合。

用户可见的虚拟天文台可视化界面，后台需要虚拟天文台系统、数据库的支持。在虚拟天文台体系中，由于所有的服务都基于标准的数据模型和服务模型，系统的数据有比较统一的交换格式，所以在进行可视化服务开发时，要注重服务的功能和与其它服务的互操作性。

2.2. 系统功能说明

2.2.1. 对不同数据库的查询

在虚拟天文台每个节点中，所包含的数据库都不尽相同。本节点中可视化的功能之一就是用户能够对所有提供服务的数据星表及星图进行选择，提交后系统能够根据用户输入的经度、纬度、查询天区的范围，返回给用户所需要的图和表。

对用户输入经度、纬度、范围的数据，要符合天文中的数据范围大小。比如说经度范围是在 0 到 360 度之间。

2.2.2. 对 FITS 图的支持

天文图像中最重要的图像格式之一就是 FITS，可视化要能够将 FITS 展现在用户面前。当然，还需要对传统格式图像的支持，如 JPEG，GIF，PNG 等等。

2.2.3. 对图像的基本操作

可视化不能仅仅是对人视觉上的展现，而且要求用户能够和图像交互，从而对图像有深的理解。比如说图像的放大、缩小，图像颜色的对比，图像亮度的对比，以从不同的方面来反映图像的本质特征。对于 FITS 图，能够根据图像上的坐标显示对应的观测像素值。

应该有小的图像窗口，这样对于视场大的天图，能够在小的图像窗口中放大图像的局部，用户可以更加清楚地看到图的属性。

2. 2. 4. 对图形的支持

在最开始的操作中，用户可以选择只查询天图或星表，或者是同时查询同一坐标的星表和天图。同时查询的目的，是为了能够在一张图上同时显示图和表。为了区别不同星表，需要在一张图上以不同的颜色画出不同星图，并且用大小不同的符号表示出来，以示区分。

2. 2. 5. 对多副图像的重叠分析

除了上述以外，用户更希望能够对不同波段在同一天区的天图，在一张图像上进行对比和叠加，以便得出有用的结论。

第3章. 可视化系统设计

3.1. 系统硬件架构

3.1.1. 网络结构图

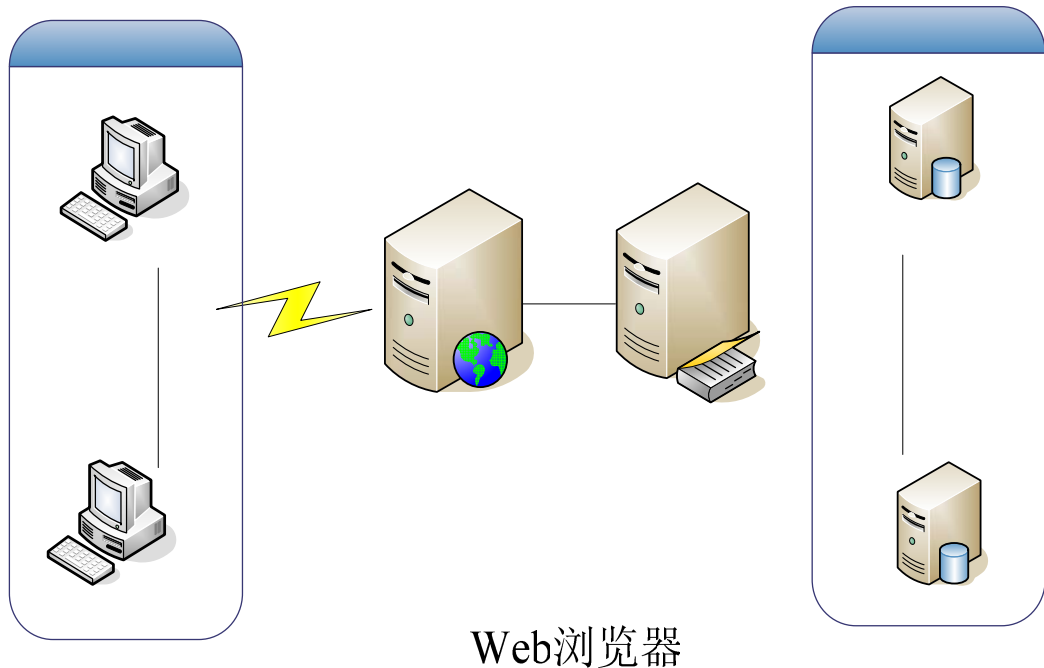


图 3-1 虚拟天文台网络结构图

3.1.2. 网络结构硬件说明

图 3-1 中的组成有三部分：

1. Web服务器(GT3^[1] Client)

Web 服务器端，用户每次通过浏览器 HTTP 访问的数据都由此提供。它的软件实现包括了多种功能，管理很多个访问用户，分别对这些用户请求进行数据处理，或者启动新的服务让用户访问需要的互连网地址，以及对出错的处理。

2. GT3^[1] Server

作为虚拟天文台的重要组成软件平台，GT3(Globus Toolkit3.x 版本)具有很多种功能。虚拟天文台是这个平台的应用。虚拟天文台将不断开发出新的服务，供用户使用。

3. DB Server(数据库服务器)

数据服务器提供了最基本的数据服务，它根据不同的数据类型，提供不同的服务，或者

下载，或者提供访问接口。可视化服务的所有数据都来源于此。目前而言，最终用户要查询的各种数据，只限于此数据服务器。

3.1.3. 网络结构图功能说明

本系统包括两大部分：

1. 数据的获取：

数据的获取主要是要通过数据网格技术来实现的，GT3 提供了网格实现的平台。数据网格中将尽量提供分布式的数据共享，数据服务有网格的后台支撑，能够取到各个网格节点的数据。在本图中，一个客户端访问某个节点，此节点直接访问本地数据库服务器。

2. 数据可视化

将获取的数据进行展现是本系统中最主要的部分之一。数据的可视化要实现一定的功能。可视化的实现硬件部分在 Web 服务器或用户终端上。

3.2. 系统设计流程

3.2.1. 数据顺序图

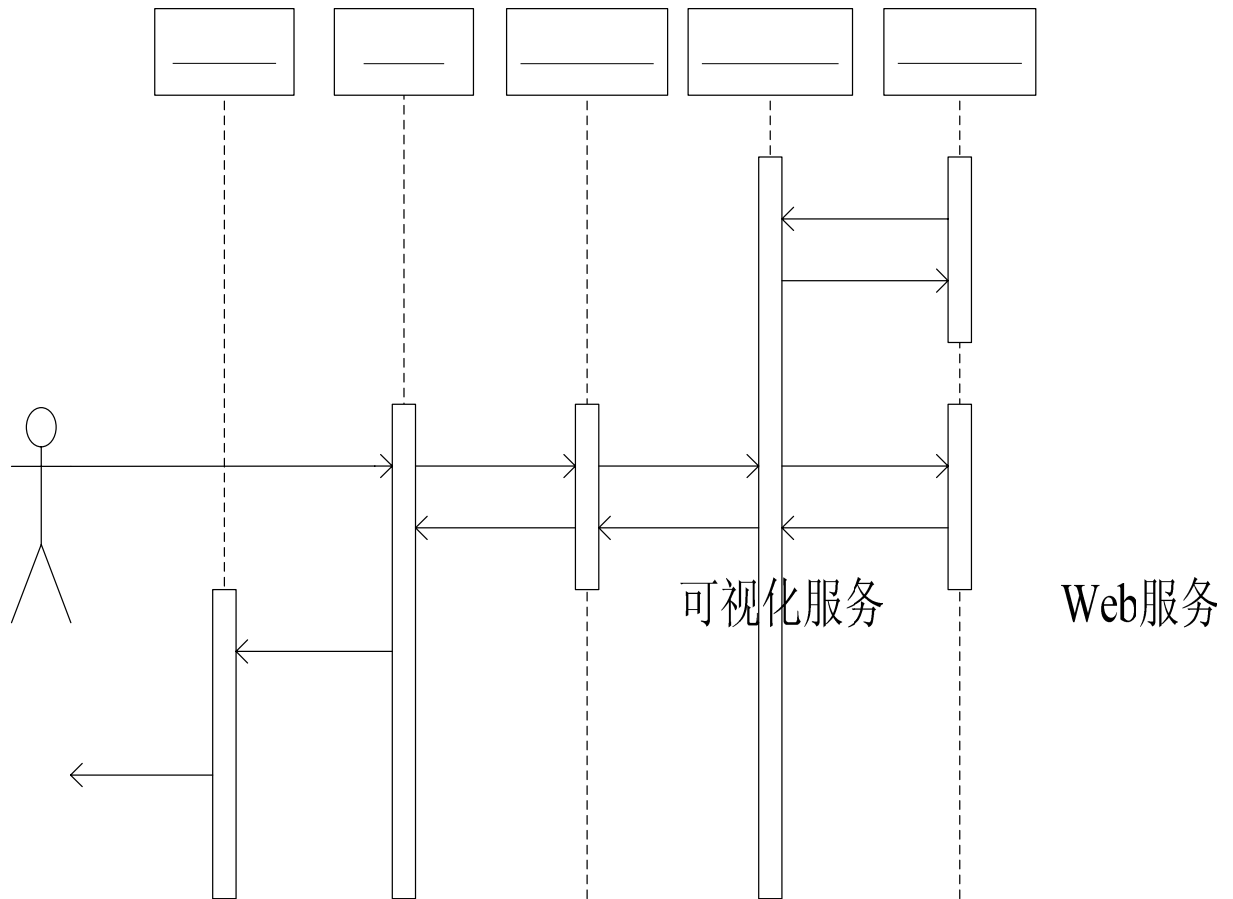


图 3-2 数据顺序图

3.2.2. 数据顺序图说明

首先，最终用户提交天文数据请求，Web 服务器上若不存在这些数据，将调用 GT3 数据服务的客户端，客户端对已注册过的数据服务提出数据请求，数据服务端将查询到的数据返回到 Web 服务器端进行处理。Web 服务器将针对最终用户的要求，将数据传给最终用户。然后最终用户对图像和星表进行操作。

请求天文数据

在操作的过程，若需要不同的星表或者图，最终可以再次向 Web 服务器提出请求，请求过程同上。

3.3. 数据服务—可视化数据来源

可视化请求数据服务。数据服务是一个网格服务，与网格技术密切相关。

3.3.1. 自动注册

任何对外提供接口的网格服务，要求已经在 GT3 容器中进行了部署和注册。

网格中服务的注册用到了 GT3 中自带的自动注册功能。

将开发出来的服务，按照 GT3 中的要求，部署到它的容器中。这个服务将通过 Java 程序或者 GWSDDL 提供服务接口。

因为GT3 部署已经有了成型的标准，按照要求的步骤部署，客户端程序就可以得到这个服务的访问点^[2]。

3.3.2. 通知框架机制

GT3 中数据服务提供了接口，供数据服务客户端访问。访问过程有不同的机制，我们可以利用其中的一种通知机制来实现访问所需要的网格服务。网格客户端可以先访问到某个网格服务节点，得到所需服务的访问点，再通过访问点对这个服务进行访问。

输入参数：经度、纬度、搜索范围、星库名。

输出参数：文件、流形式，字符串形式的 VOTable 星表。

操作:根据具体的需要定义。

3.4. 星表数据展现(XML 数据到图形显示)

在虚拟天文台中，为了共享数据，需要一种标准的格式，能够在网格系统中进行数据的传送和互操作。为了将数据说明和数据内容分开，查询到的星表数据最好都是 XML 格式的。

对于这些数据，Web 服务器需要提供如下的功能：

1. 提供下载；
2. 转换给最终用户需要的处理格式；
3. 能够通过解析在浏览器中可视化为二维的。

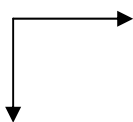
具有标准 XML 格式的 VOTable 作为天文的标准格式，得到了虚拟天文台的技术实现的认可。VOTable 为天文的网格技术实现，特别是统一数据访问，提供了很好的数据格式标准。这也是虚拟天文台做为最好的数据网格试验床平台的原因之一。

3.5. 星图及表的数据展现功能

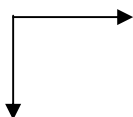
3.5.1. 坐标系统功能说明

直角坐标矩阵转换，是当代天体测量学计算恒星位置的通用方法。这种方法在数学上是严格的。随着计算机的普及，直角坐标矩阵转换的计算在时间上没有任何问题。对于图像的显示，涉及到的坐标有很多，例如一般的用户坐标，还有天文上用到的世界坐标，画图的画布坐标等等。这些坐标之间的转换是实现显示图像、对图像进行处理、以及画图的基础，所以首先我们就要考虑各种坐标之间的转换关系。

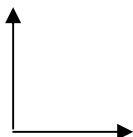
显示器坐标^[3]：起始点一直是去最左上角。所有的像素都计算在内，但是变换不在考虑之列。



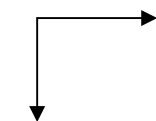
画布坐标：起始点在图像的最左上角。所有的像素都计算在内，但是变换不在考虑之列。



图像坐标：当所有的变换都没有变换之后，起始点在左下角(FITS 格式)。图像坐标与 FITS 图像的坐标相对应。



用户坐标：起始点在最左(0.0, 0.0)，所有的转换没完成之后。用户坐标和图像坐标类似。



世界坐标系^[4]：世界坐标是从图像坐标转换得来，如果可以从图像坐标中得到关键字。

望远镜坐标：一套精心的 FITS 规范被定义用来指定物理或者世界，坐标被附在 N 维图像的每个像素后面。世界坐标系说明坐标用于在某些多维参数空间中定位一个尺度。在一般情况下，通过天图中经纬度与平面图像的像素建立相关性最方便。

为了得到天体位置的大概位置，将世界坐标系与图像相拟合并不麻烦。当拟合一个WCS^[5]时，一个名义上的WCS能够从通常在望远镜图像中代表的关键字中创建出来。所以假设望远镜由RA和DEC关键字代表的所指方向是在图像的中心是合理的(可能不准确)，图像的坐标是 FITS图像中的NAXIS1/2, NAXIS2/2(假设旋转是零度, CROTA1 和CROTA2 也被设置为 0)。

FITS 图的处理是天文图像处理的重点之一。其中对图像的影射和操作与 WCS 坐标的转换之间的关系是主要的内容。

几十种映射方法支持 FITS 标准。例如正切映射、正弦映射、等方位角映射、墨卡托映射、圆柱等面积映射、圆锥透视映射等等。

3.5.2. FITS I/O

技术上对 FITS 的支持，首先要支持对 FITS 的读写操作。

这个组件应该可以用来读写FITS格式的图像，包括很多HDU、二进制和ASCII表，以及图像的扩展^[6]。读和写压缩的图像也应该被支持。这个组件不会实现压缩支持，但是会利用已有的类。VisAD^[7]包包括我们必须的看起来对FITS很好的支持。这个组件不能处理图像的显示，只能读写FITS文件。

在 FITS I/O 组件中，实现了从文件、URL 或者流中来访问 FITS 数据。

3.5.3. JAI FITS 支持

其次，我们要增加对 FITS 的处理操作，在提供的对外接口中，要做到与图像格式无关。这样其他组件在对图像进行处理的时候，只需要在初始化的时候说明是 FITS 图像，在以后的图像操作中，程序能够一视同仁而不是针对具体的图像格式进行处理。

这些功能的实现，可以借助于 SUN 公司开发的 JAI 图像处理包。在 JAI 包中增加对 FITS 图像的支持，这个组件中类的基本结构将通过 JAI 的接口表现出来。

这个组件将相当复杂，因为它需要处理不同的颜色模型、颜色图，和各种图像数据类型。它最后应该也支持真彩的 FITS 图像，这些真彩的图像由红、绿、蓝图像组成(NAXIS=3)。

另外一个重要的因素是支持大的图像，与 8k*8k 浮点在一个量级上。JAI 接口可以根据需要装载分离的图像小片，从而以很少的内存更快地显示数据量大的图像。

JAI 提供了一种接口，可以访问图像属性。此包可以用来提供通用的 FITS 关键字访问。

Java2D 提供了基本的接口来管理坐标和变换，比如说旋转和缩放。这个组件应该只需要直接处理图像坐标的横坐标 X 和纵坐标 Y。其他的坐标系统，例如世界坐标系统，必须通过其他不同的组件来实现。

尽管上述的 FITS I/O 包定义的很好，有简单的接口，但是要配合 JAI 仍然要做一些修改提高性能。

一旦我们有基本的 FITS I/O 类，就会非常容易地 FITS 文件加入 JAI 中。

```
ImageCodec.registerCodec(new FITSCodec());
```

```
...
```

```
PlanarImage im = JAI.create("fileload", "myfile.fits");[8]
```


3.5.4. 支持其他图像格式

Java2D 和 JAI 的架构设计很好，在这一架构下以相同的方式来处理不同格式的图像是可行的，所以应用程序可能不知道也不关心实际的图像格式是什么。

JAI包提供了接口，这个接口可以增加新的图像类型，并增加图像处理的操作到操作注册器中。下图就说明了JAI一般怎样能够用来处理不同图像类型^[4]：

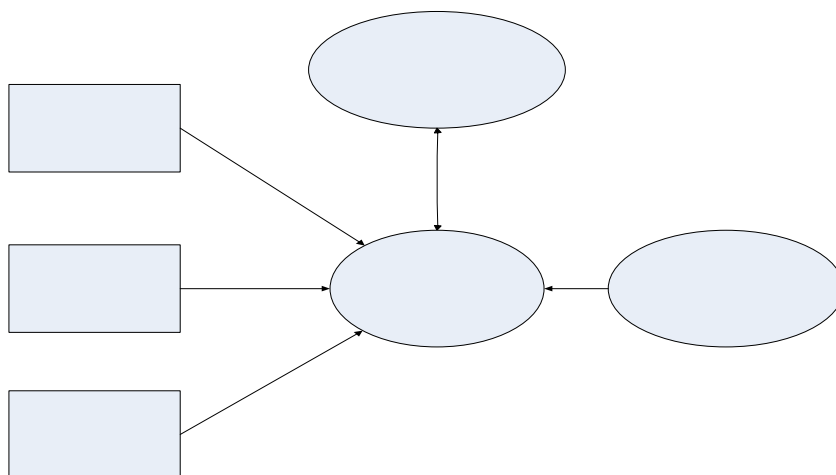


图 3-3 JAI 图像处理和显示

图像类型

3.5.5. 图像处理操作

这是一个独立的组件，使用 JAI 接口增加天文相关的图像处理特征。它独立于任何特定的图像格式。

FITS

3.5.6. 图像显示小窗口

这个组件实现了一个图像显示的小窗口，这个小窗口有可选的滚动条，需要时可以使用。这个组件独立于任何特定的图像格式。上面谈到的 JAI 包可以自动判断图像类型，因此图像类型就从应用程序中“掩藏”起来了。

JPEG

如果在 JAI 包中增加了对 FITS 的支持，那么在装载 FITS 文件时也可以自动识别。

图像显示小窗口也会提供一些方法，这样用户根据指定的坐标系统，能够方便地在图像上画图或者访问图像数据。

3.5.7. 图像 Zoom 小窗口

GIF

图像 Zoom 小窗口是图像显示组件的特殊版本，用户可以通过鼠标事件来放大显示图像的某个部分，对鼠标事件的控制能够改变 Zoom 的放大倍数。

3.5.8. 图像面板小窗口

图像面板小窗口用一个矩形框显示一副大图像中小视角范围的图像内容。用户能够用鼠标拖动矩形窗口来控制大图像的视图范围。

在处理大的图像时不是将整个图像装载进内存，这里的图像处理需要得到特别关注。

3.5.9. 图像图标

这是一个简单的组件，图标从一个大的图像中的来。这样能够用来更加容易地定位关闭的图像显示窗口。

上述所有的图像组件应该连接在一起，这样主要图像中进行水平剪切或者改变颜色或者改变方向时，这些改变能够通知到其他图像视图中。具体的实现可以用 Observer/Observable 模式，当有任何改变，图像生产者通知图像消费者。

3.5.10. 图像控制面板

除了上面提到的简单图像组件，还有更加复杂的、高级的组件组合。这些组件有灵活实用的垂直或者水平的布局。组件的组合见表格 3-1。

图像控制面板就是这样一个组件，它包括下面任意几个：图像面板小窗口，zoom 小窗口；还包括一些标准的图像控制器和标签。详细见表 3-1。

这样的面板粘贴到图像显示窗口中去，可以控制图像的视图，包括方向、放大倍数、可见区域。生成这个面板的组件应该是 Java Beans，这样用户可以很容易地用接口来设计新的面板。

基本图像组件	组件或小器件	图像控制面板窗口	主要图像显示窗口	选择星体窗口
	FITS I/O			
	JAI FITS 及其他图像格式支持			
	WCS			
	图像处理操作			
	图像显示小窗口		1	
	图像 Zoom 小窗口	N(>=0)		1
	图像面板小窗口	N(>=0)		
	图像图标			
小器件	图像控制器	N(>=1)		
	标签	N(>=1)		
复杂图像组件	图像控制面板		1	
	信息面板			1
	HDU 显示小窗口			

	图像颜色管理器			
关于像素的操作	图像剪切			
	像素表			
图形处理	图像作图			
	图像作图编辑			

表格 3-1 星图及表的数据展现组件要求

3.5.11. 主要图像显示小窗口

这个组件正是一个图像显示小窗口和控制面板的结合。将它作为最上级的窗口并增加一个菜单栏，就构成了一个可以运转的应用程序。

3.5.12. 选择星体小窗口

这个组件和图像显示小窗口联在一起，用户就能选择图像中的星体。它用质心算法来定位所选星体的中心，然后显示星体的信息和统计数据。这个小窗口将一个可配置的 zoom 小窗口和一个可配置的信息面板结合在一起，显示鼠标周围的区域信息。

在许多实际操作中，另外的组件甚至应用程序，会要求用户在图像中选择一个点或者天体，再基于所选择的对象来完成某些操作。

3.5.13. HDU 显示小窗口

当一个图像文件包括多副图像(比如说多个 FITS HDU)，这个组件能够用来给用户一个提示，选择需要的图像和表。这个组件应该包括一个表，罗列出图像文件内容，或者这个文件中图像的一部分的信息。

在某些实例中，用户可能希望将不同图像拼凑成一副完整的图像，比如说当这些图像从不同的探测器中观测得来的。

除了显示图像的组件，许多支持的组件需要用来管理图像颜色和进行图像处理操作。

3.5.14. 图像颜色管理器

图像颜色管理组件应该提供一个窗口接口，这样用户可以对给定的图像选择一个颜色图，设置任意的选项来处理图像颜色，例如有多少种颜色能够用来显示图像、怎样进行颜色调配。这个管理器也应该可以在某个范围内面或外边，对坏的像素指定特定的颜色。这个组件也允许用户指定怎样解释真彩图像，比如说 NAXIS=3 的 FITS 图像。

3.5.15. 图像处理工具

Java2D 和 JAI 包包含了很多对图像的操作，我们可以充分利用这些功能，除此以外，还要增加与天文图像相关的特殊处理。这个组件应该包括一个控制器，让用户可以访问所有和图像处理相关的特征。这些特征包括：操作给定图像的亮度、锐化度和对比度，颜色的调配，登记，指数、柱状图均衡化，旋转，放大，图像算法等。

3.5.16. 图像剪切和像素表

这两个组件是用户在像素级别上，提供给用户所选图像区域的信息。

图像剪切是允许用户在图像上拖动一根贯穿图像的线，显示沿着这线的像素值。

像素表是在一个表中显示鼠标所指点周围的像素值。

3.5.17. 图像作图

这个组件提供了一个接口来在图像上作图。它应该支持很多标准的图形和符号各种坐标系。

3.5.18. 图像作图编辑

这个组件提供了一个编辑器，用来在图像上画图和操作作图。它应该可以保存作图，或者与图像一起保存在 FITS 表中，或者在一个单独的文件里。当用户已经完成或者是取消一个画图操作，它应该也可以将操作步骤通知给其他的组件。

3.6. 客户端和服务端功能

3.6.1. 程序存放端的选择

对于 Web 程序，可以将上述各种图像操作的功能放在服务器端，也可以放在客户端。

方式一：将程序放在客户端。缺点是要初始化程序，速度比较慢，但是能够提供给用户强大的互操作功能。

方式二：将程序放在服务器端。此种方式将数据都放在服务器端，优点是服务器功能比较强大，缺点是如果与用户交互较多不能提供很多操作功能。如果只是简单的操作，将软件程序放在服务器端是不错的选择。

从与用户交互角度而言，选择上述第一种方式。

3.6.2. 服务器端处理内容

服务器端的程序，部署后才能给外界提供服务。服务器端的处理内容主要包括如下几个方面。

1. 和虚拟天文台数据服务客户端相衔接。

编写虚拟天文端数据服务客户端程序，能够传递参数给虚拟天文台数据服务服务器端，能够接收到该网格服务传回来的数据。

本章 3.3 节中已对此做了描述和设计。

2. 对从数据服务传回来的数据进行处理。处理的内容包括：

- a. 对图像数据，可以直接放在服务器端，存成临时文件；或者将需要的少量数据不经过重新的包装和处理直接传送给最终用户。

- b. 对表的数据，提供实时下载，或以 Web 页面显示出来，或者解析传送给客户端。

要求：服务器端能够将数据迅速传给客户端，超过一定的时间将不再等待。

3. 能够将最终用户需要的数据传给客户端。

数据的传输要保证正确、完整，这项要求可以通过序列化等技术来保证。

3.6.3. 客户端功能

客户端包括有本章 3.5 节中提到的所有星图和星表的数据展现功能。

从用户的角度而言，能够实时地操作提供的各种功能。

当然程序对用户的操作范围也要加以一定的限制，来保证程序实现的可行性。例如对天区的数据查询需要加以一定的限制，1 度天区是比较合适的。对于超时(例如一分钟)还未传到的数据，能够取消这个查询的操作。

与 Web 服务器端通信，实时取得数据的功能。

能够配合 Web 服务器端接收数据，让程序能够方便地处理。

参 考 文 献

- [1] Globus Toolkit. <http://www-unix.globus.org/toolkit/>
- [2] S.Tuecke, K.Czajkowski, et al. Open Grid Services Infrastructure. 2003-04-05.

- [3] JSky Image Related Components. http://archive.eso.org/JSky/draft/image_comp.html
- [4] FITS World Coordinate System(WCS)
http://fits.gsfc.nasa.gov/fits_wcs.html
- [5] World Coordinate Systems. <http://tdc-www.harvard.edu/software/wcstools/wcstools.wcs.html#FITS>
- [6] Greisen, E. W., Calabretta, et al. Astronomy & Astrophysics, 2002(395): 1061-1075
- [7] [VisAD] <http://www.ssec.wisc.edu/~billh/visad.html>
- [8] Implementing FITS image support for JAI.
<http://archive.eso.org/JSky/draft/fitsjai.html>

第4章. 可视化服务的实现及关键技术

4.1. 系统说明

4.1.1. 开发工具的选择

各国针对虚拟天文台的开发工具不尽相同。中国虚拟天文台尽量采用了一些开发源码的软件。这样不仅有利于在有些方面进行修改，也减少了开发成本。

网格平台是 Globus Toolkit，现在最新的稳定版本是 3.2。在开发过程中，我们从最初的 3.0 Alpha 版本用起。随着它不断地更新和增加新的内容，Globus Toolkit 提供的数据库管理、传输、安全等方面的功能亦日趋完善。

Globus Toolkit是用 Java开发的工具包，虚拟天文台对它的选择决定了我们的开发语言Java。JDK中包含的Java标准包包含有我们需要开发图像、图形的类，JAI^[1]包中有需要的图像处理扩展功能的类。

在Web服务器的选择上，作为处理Java服务器端程序的Tomcat^[2]比较合适，JSP^[3]脚本也被采用了。

Java程序开发工具有用到的是Eclipse2.1^[4]，Ant1.5^[5]。编辑工具可以直接在Eclipse上开发。

选择 Java 的原因

1. 因为我们的需求是显示天文图像和制作表格数据，Java2D 包中有对图形的支持，JAI(Java Advanced Imaging)包中有对图像的支持，它们分别提供了强大的功能，例如直方图均衡化，颜色的调配，设置图像剪切线。

2. 对用户界面的支持。

Java 基本组件 Swing 工具提供了用户界面需要的一般小窗口部件。作为 Java 标准的部件，Swing 中除了一般的按钮和菜单集合以外，还提供了丰富的用户界面组件，例如表、树等小部件。因为是标准的 JDK 中的一部分，Java Applet 在用到这些组件时不需要下载许多额外的代码。

4. 1. 2. 关键技术的说明

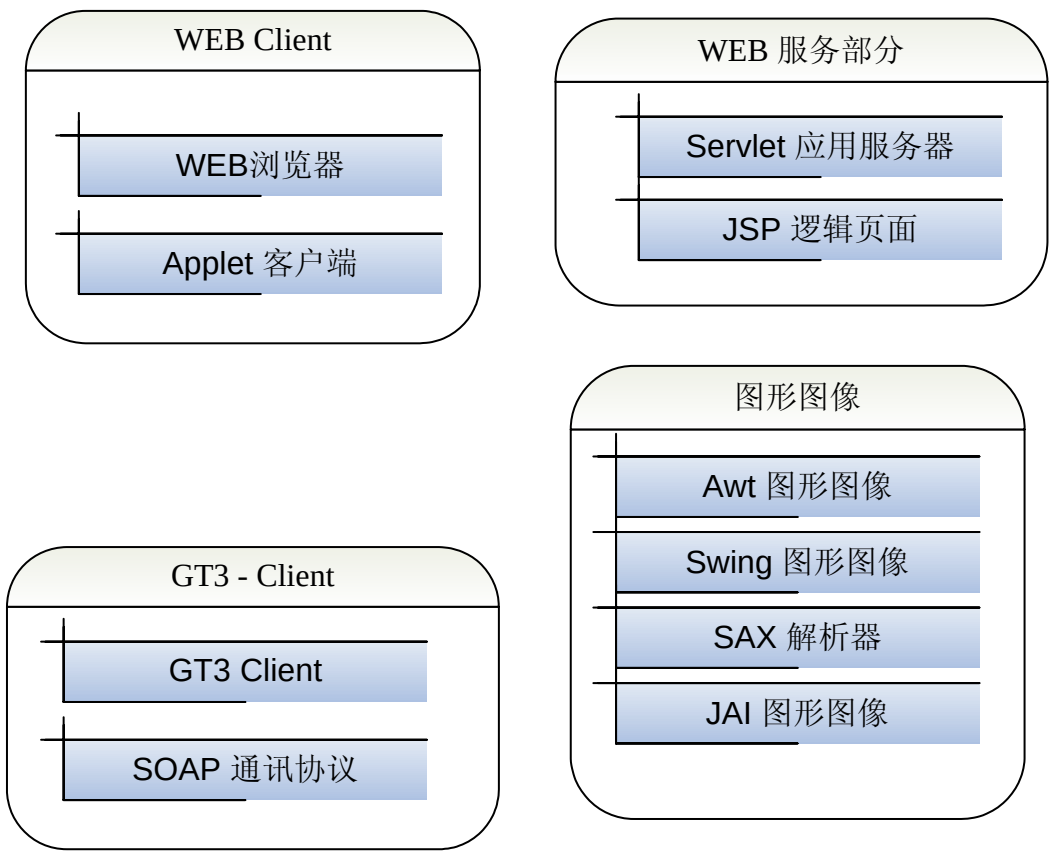


图 4-1 各部分关键技术说明

图 4-1 说明了各个具体实现中需要用到的技术。

4. 2. 软件架构

4. 2. 1. 数据流图

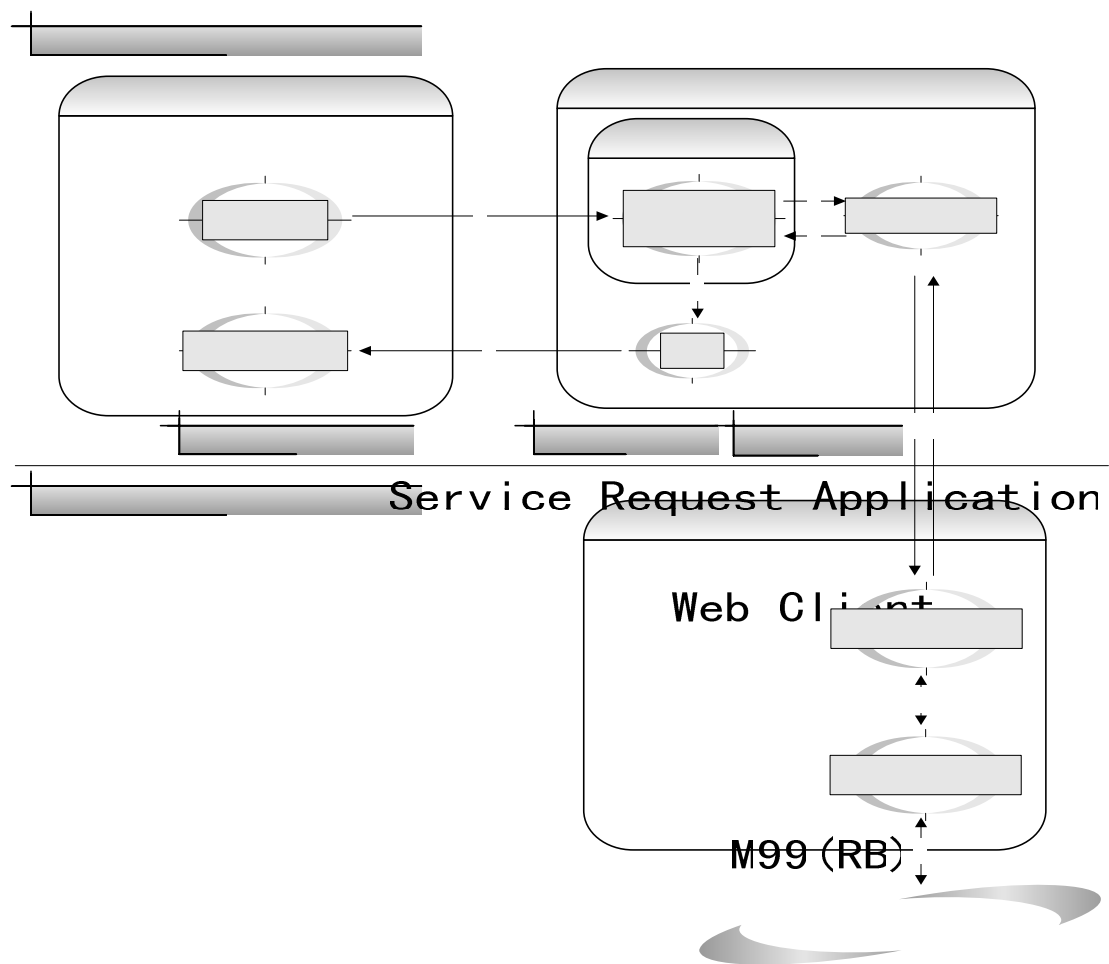


图 4-2 可视化数据流图

前提:

3D Picture

9

- 1. 所有的实现都由用户触发，服务器端负责转发处理数据，对处理过的数据将按其访问时间做定时的清理。
- 2. 对虚拟天文台数据服务端，没有返回数据的时候将取消用户的操作。

图 4-2 流程说明:

- 1. 首先，用户访问包含可视化功能的 Applet 的 HTML 页面，用户所有的操作都在这个 Applet 上进行。第一步，用户输入所需要查询的星表及图的参数；
- 2. Servlet 接收到 Applet 传递过来的参数，触发某个事件或者调用另外一个 Servlet，从而访问虚拟天文台数据服务客户端；

3. 虚拟天文台数据服务客户端通过SOAP^[6]协议访问数据服务服务器端, 调用相应的程序(图中第4步), 访问到数据(图中第5步);

6. 数据服务器端返回给数据服务客户端查询的 VOTable 格式数据;

7. Servlet 取得数据服务客户端数据, 并进行临时文件存档;

8. Servlet 端用 SAX 方法对临时文件进行解析, 通过 java.net 包中的方法返回给 Applet, 或者直接对用户提供下载。Applet 端接收数据的程序如下:

```
URL fitsurl= new URL("http://localhost:8080/"+"/servlet/FitsServ?" +
path);
```

```
URLConnection fitscon = fitsurl.openConnection();
```

9. 若数据解析正确, Servlet 将解析后的数据返回给 Applet 客户端, 以便进行对图像和表的各种操作; 若数据解析出错, 会返回 JSP 出错页面。

4.3. Globus Toolkit 的应用实现

虚拟天文台可视化的数据来源于底层的 Globus Toolkit 中的数据服务。要从数据服务中得到数据, 需要访问数据服务接口。

4.3.1. 网格服务接口描述

网格服务的对外接口提供了两种实现的方法, 它提供了一定的规范, 能够被客户端的程序访问。

1. Java 或 IDL 接口通过 Java2wsdl 工具来转换:

优点是非常简单, 一个 10 行的 Java 接口很容易就转换成 50-100 行的文档。

2. 直接用GWSDL^[7]语言写。它的优点是:

a. 语义上更加丰富:用 WSDL 比用接口语言更加具有“表达性”。有一些操作或者要求用 Java 不能表达出来, 但是可以用 WSDL 表达出来。

b. 更多的控制:从接口语言开始意味着我们不得不执行转换程序把 Java 代码变成 WSDL 代码。尽管把设计实现的工作变的更加容易了, 但是每次在软件或硬件之上增加一个新的抽象层, 程序员就会对最后的结果失去一些控制。这就意味着最后的 GWSDL 代码很有可能不是设计人员精确想要的。

GWSDL 在网格服务中发挥了重要作用, 它最突出的两个优点在于:

1. PortType 继承

2. ServiceData

说明: GWSDL现阶段是WSDL的扩展, 有一些WSDL1.1 没有的特征。Global Grid Forum^[8]已经表示, 一旦WSDL1.2 成为W3C稳定的标准(Recommendation , a stable

standard), 它将会以 WSDL1.2 替代 GWSDL^[9]。

4.3.2. 实现的基本步骤及部署自动注册

上面只是谈到了服务接口的描述情况。要实现一个网格服务, 总共需要四项:

1. 一个 service 接口,
2. 一个 WSDL 文件,
3. 一串 stub 文件,
4. 一个服务的实现。

有了这些文件之后, 可以通过 WSDD(Web Service Deployment Service) 格式写的部署描述器 deployment descriptor, 将网格服务部署到 Web 容器中。这个 Web 容器可以是 GT3 自带的, 也可以是 Tomcat。

所有实现的步骤, 都可以通过 ant 工具来实现, 如图 4-3^[9]。

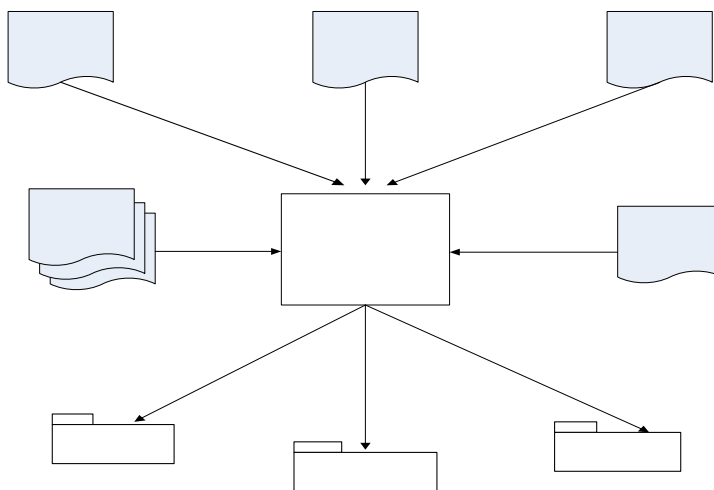


图 4-3 一个网格服务所需要的全部包和文件

4.3.3. Globus Toolkit3 提供的两种实现方法

对于已经定义好的用户接口, 欲利用 Globus Toolkit 核心提供的资源管理等各种功能, 则需要实现这个用户定义好的接口。实现方式有两种

1. 通过继承来实现

VOTable.java *VOTable*
服务接口(Java) *服务实现*

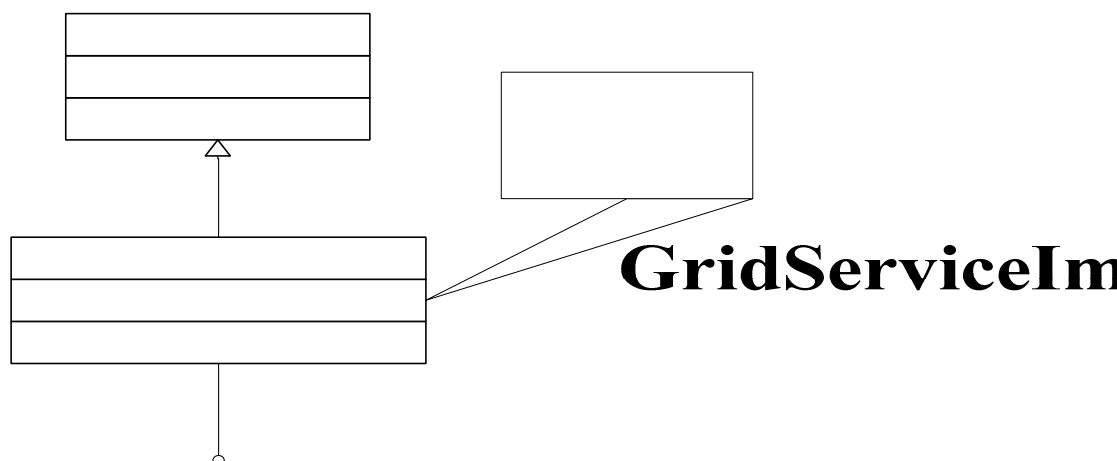


图 4-4 通过继承的网格服务实现

2. 通过代理实现(Operation Providers)

代理方式, 允许我们将PortType中的操作分布在几个类中。比如说, 它将以前的MathImpl分到三个类中: 一个为数学计算操作, 一个是三角函数操作, 一个是矩阵操作(operation), 如图 4-5^[9]。这些类的每一个类都叫做操作提供者。

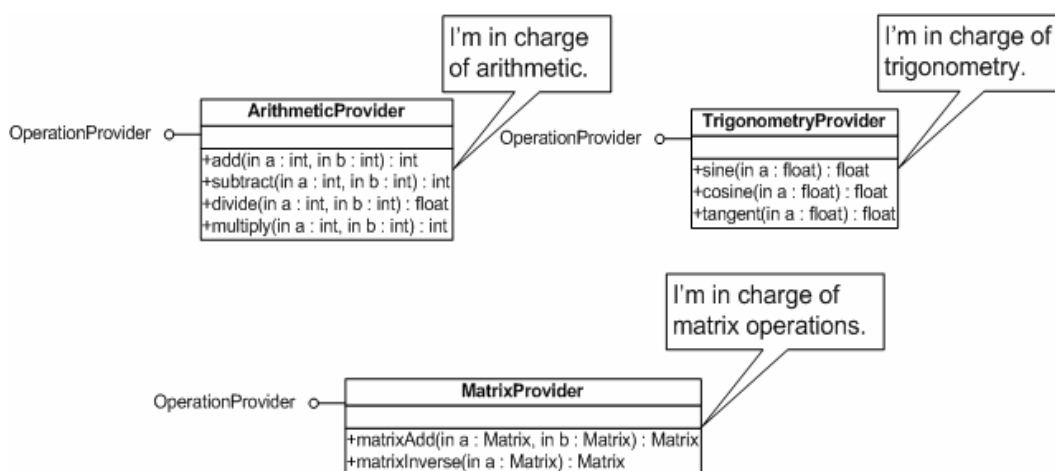


图 4-5 通过代理的网格服务实现

注意到这些类没有扩展(extend)任何基类。它们仅仅需要实现一个叫做OperationProvider的接口。

OperationProvider接口被包含在Grid Service容器中, GridServiceImpl在容器中提供了基本的Grid服务功能, 所以在图 4-5 中尽管没有如图 4-4 扩展任何基类, 但是仍然实现了网格服务的核心功能。

继承方法与代理方法比较:

由于类的多继承在Java中不允许, 而网格服务实现类已经继承了GridServiceImpl类, 在这个时候如果程序员希望继承自己写的其它类就行不通

了。代理方法在这种情况下就显示出了比继承方法的优越性。

4.3.4. 日志

GT3 调试信息是基于 Apache Jakarta Commons Logging 组件。

Apache Jakarta Commons 项目的目的是开发可以重用的 Java 组件，比如说有效的类，命令行操作解析等等。这个项目的组件之一是 Commons Logging 组件，它允许我们很容易的从 Java 类生成日志信息。

Commons Logging 的日志有六个级别：

1. 调试(debug)
2. 跟踪(Trace)
3. 信息(Info)
4. 警告(Warn)
5. 错误(Error)
6. 致命错误(Fatal)

过滤操作可以采取如下几种：

1. 所有的或调试
2. 跟踪
3. 信息
4. 警告
5. 错误
6. 致命错误
7. 关闭

4.3.5. 服务数据(ServiceData)

网格服务结构中重要的一部分是服务发现，服务发现允许我们发现满足我们需要的网格服务的 URI。

服务数据是和一個网格服务相关的结构性信息的集合。这样的信息必须十分方便查询，因此网格服务能够根据他们的特征被分类、索引。总之，服务数据是服务发现的核心部分，它与通知(notification)紧密相关。

任何一个网格服务实例都有服务数据和它相关。图 4-6^[9]显示出了 Service Data 在 OGSA 中的结构：

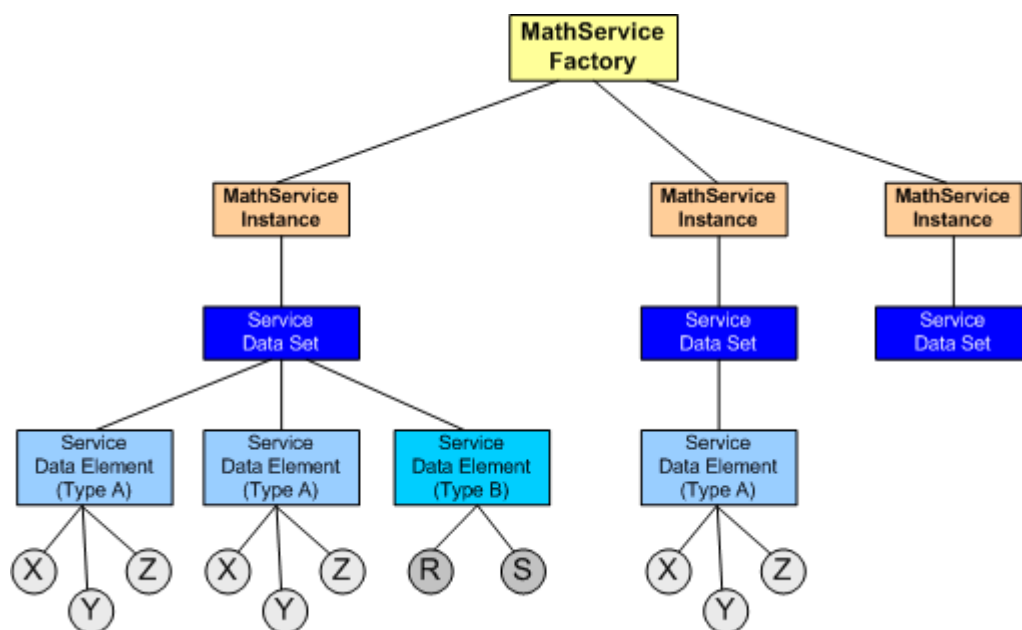


图 4-6 服务数据 (Service Data) 在 OGSA 中的结构

我们可以注意到每个实例有个服务数据集合 (Service Data Set)。一个服务数据集合能够包括 0 个或者多个服务数据元素 (Service Data Element, 简称 SDE)。每个 SDE 可以是不同的类型。这样网格平台可以根据用户的具体需求, 选择最相匹配的网格服务并将服务访问点提供给用户。

在此, 接口描述语言 GWSDL 的其中一个优点—服务数据就体现出来了。

GWSDL 允许我们指定一个网格服务中的服务数据, 比如说一个网格服务能够包括的一种类型的 SDE 的最小和最大值。

4.3.6. 通知框架机制

通知机制并没有任何新内容, 它已频繁的使用在软件设计模式中, 有着其他不同名称: Observer/Observable, Model-View-Controller (J2EE 的 MVC 模式)。

假设软件有几个不同的部分 (例如 GUI 和应用逻辑, 客户端与服务端等等。) 并且软件中的一部分需要注意软件中另一部分的变化。例如 GUI 需要知道在数据库中内容什么时候变化, 就可以立刻将新内容显示给用户。在 C/S (客户端/服务器端) 模式中很容易做到这点: 假设客户端知道服务端什么时候达到了特殊的状态后, 以至于这个客户端可以及时的对服务端执行一系列调用。

使客户得到通知的最原始方法就是投票方法, 即我们通常所说的轮询法。客户端周期性的访问给服务端, 问服务端有没有任何变化。

例如, 让我们假设有一个服务端, 很多客户端需要在这个服务端签证以获取指定的新消息。一旦已签证的客户端的数目达到要求后就发送新消息, 但是由另一个服务器发送新消息的 e-mails。这另一个服务需要知道什么时候有足够的已签证

的客户端，以至于它能发送 e-mails 了。

第一个服务器叫 Observable，第二个服务器叫 Observer。

一旦发生变化 Observable 就会立刻通知我们，然后我们可以有根据的进行动作。这就是通知方法。在使用 Observer/Observable 设计模式时有两种方法：Pull 方法和 Push 方法。每个方法都有它的缺点和优点。

Pull

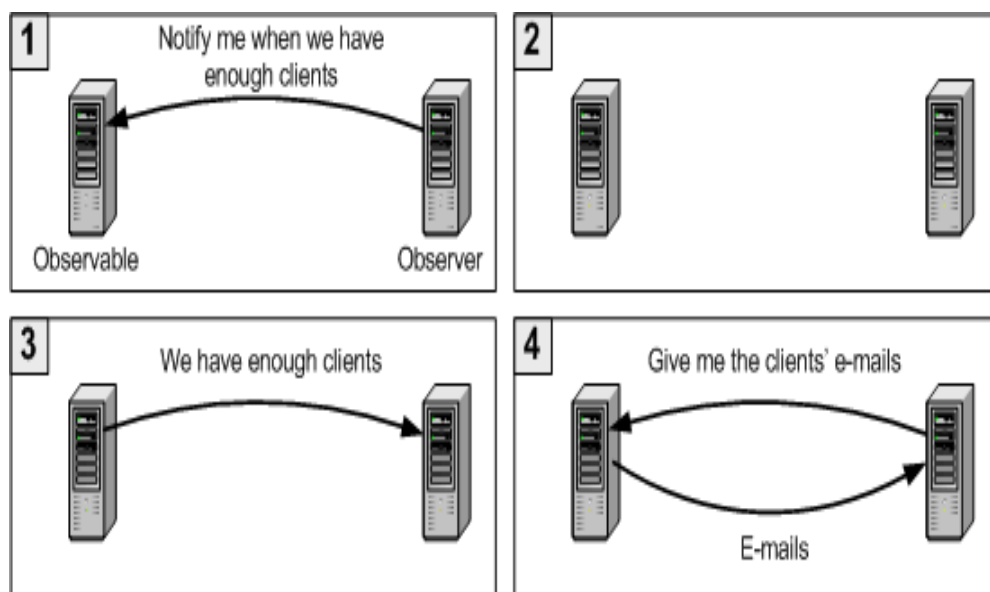


图 4-7 Observer/Observable Pull 模式

图 4-7^[9]所显示的就是Pull方法，说明如下：

1. Observer 要求 Observable 一旦有足够的客户端就通知它。Observable 保存了所有在它这里注册的 Observer。这步通常被称为订阅或者注册步骤。

2. Observer 和 Observable 各自做自己的事情。到目前为止还没有足够的客户端。

3. 已经有足够的已签证客户端了。Observable 通知它所有的 Observer(不只一个 Observer)已经有足够的客户端了。

4. Observer 要求 Observable 发送 e-mail 地址。

这个方法效率很高。在这个简单例子中，与投票方法相比网络通信压力减少了至少一半。

在严格实现过程中，Observable 简单的告诉 Observer “变化” 发生了。这个通知可以说明确变化的类型(例：“We have enough clients”)但没有包含与这个变化相关的信息。注意，在通知之后，Observer 必须再向 Observable 执行一个调用以获得 e-mail。这个可能看起来是多余的调用，但是这种方法在下面这两种情况很有用：

1. 一旦变化发生后每个 Observer 从 Observable 得到的信息不同。

2. 当一个通知不是意味着 Observer 将从 Observable 请求数据时(Observer 可能选择忽略通知)。

Push

在这方法中, 我们允许数据与通知一起传输。在我们的例子中, 通知将包含 e-mail 地址。这种方法在下列情况下很有用:

一旦变化发生每个 Observer 获得相同的信息。

一般, Pull 方法赋予客户端在收到通知后更多的权利获得它需要的数据。虽然 Push 方法对客户端能收到什么信息做了限制, 但由于我们节省了与 Observable 额外的通信开销, 这个方法效率更高。

GT3 中的通知机制

在 GT3 中通知机制是与服务信息密切相关的。事实上 Observer 不向实例(Instance)订阅, 而是向实例中的一个特定的 SDE 订阅。图 4-8 演示了拥有一个单独 SDE 的 MathService 实例, 和几个客户端如何完成订阅工作的。

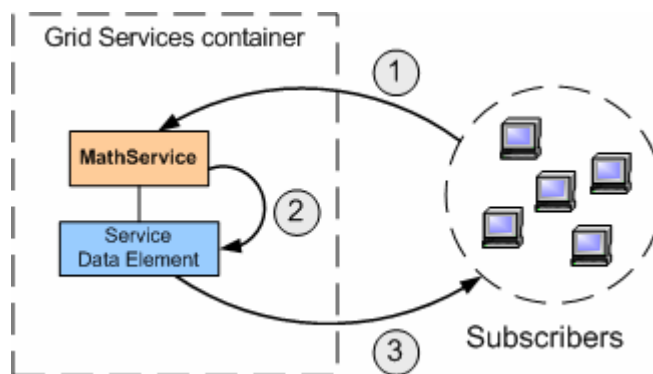


图 4-8 GT3 通知订阅

addListener: 这个调用是向特定的 SDE 中添加要求订阅的客户端(在调用过程中详细说明)。

notifyChange: 无论什么时候发生变化, MathService 实例要求 SDE 通知它的订阅者。

deliverNotification: SDE 通知订阅者发生变化。

通知允许客户端被通知当网格服务发生了变化。

在测试的时候, 我们需要创建实例, 这样客户端可以订阅。我们不能用瞬间(transient)实例, 因为这样的实例就是为了避免在客户端间数据共享。但这并不是说在瞬间实例我们就不能用到通知模式了: 一个客户端可能用到瞬间实例几个小时, 尽管这个客户端是唯一的订阅者。它可能需要收到通知, 当网格服务被第三方面的系统改变, 而不是网格服务客户端。

4.3.7. 数据服务的初步实现

虚拟天文台可视化的数据来源于底层的 Globus Toolkit 中的数据服务。

要从数据服务中得到数据, 需要访问数据服务接口。接口可以是 Java 的接口, 或者是由 gsdl 描述的接口内容。

对于 Java 写的 Interface 程序，我们只需要通过 rmi 的方式调用其中的方法。对于虚拟天文台数据服务端，尽管提供的仅仅是 GWSDL 描述的接口。但是访问点仍然是 Java 程序，说明如图 4-9。所以对数据服务的客户端而言，只需要理解所提供的 GWSDL 文件。

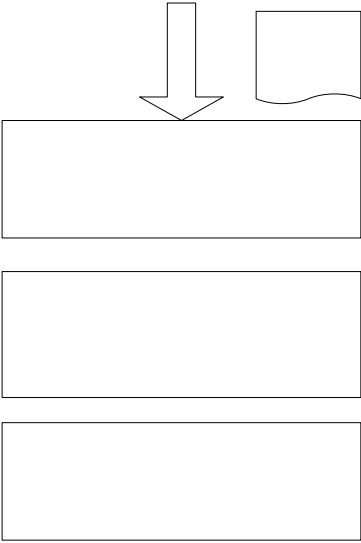


图 4-9 接口流程

GT3 中数据服务提供了接口，以便客户端可以访问。访问过程有不同的机制，我们可以利用其中的一种通知机制来实现访问所需要的网格服务。网格客户端可以先访问到某个网格服务节点，得到所需服务的访问点，再通过访问点对这个服务进行访问。如图 4-10:

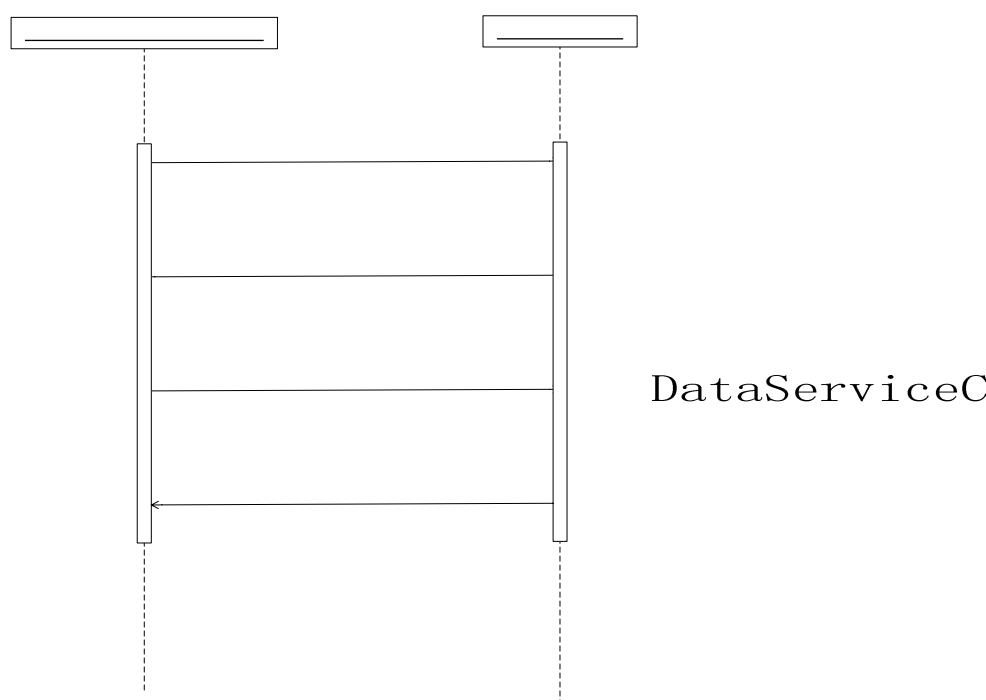


图 4-10 数据服务客户端与服务器端交互顺序图

数据服务中客户端与服务器端的交互是扩展了java.rmi包实现的，在机制上应用了上节中的Pull机制。在服务器端和客户端，程序的模型实现如图 4-11，图 4-12^[10]。

在类的实现上有

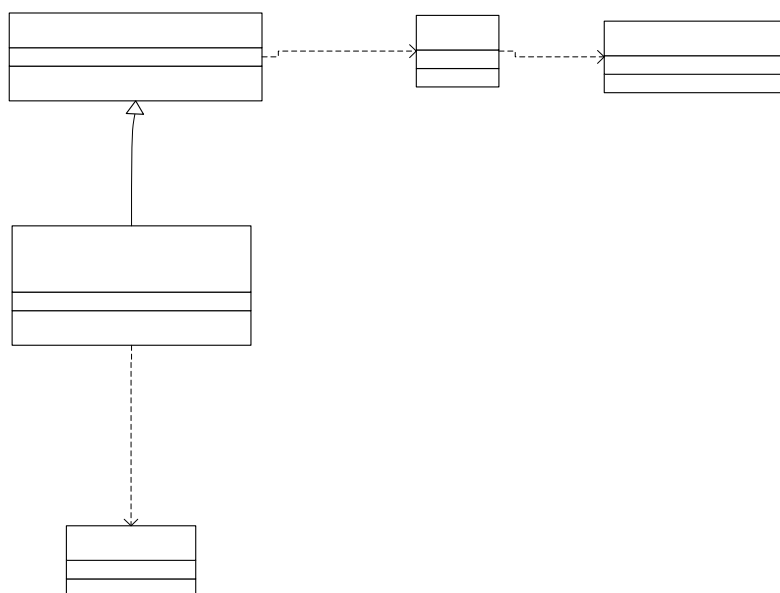


图 4-11 网格服务客户端程序模型

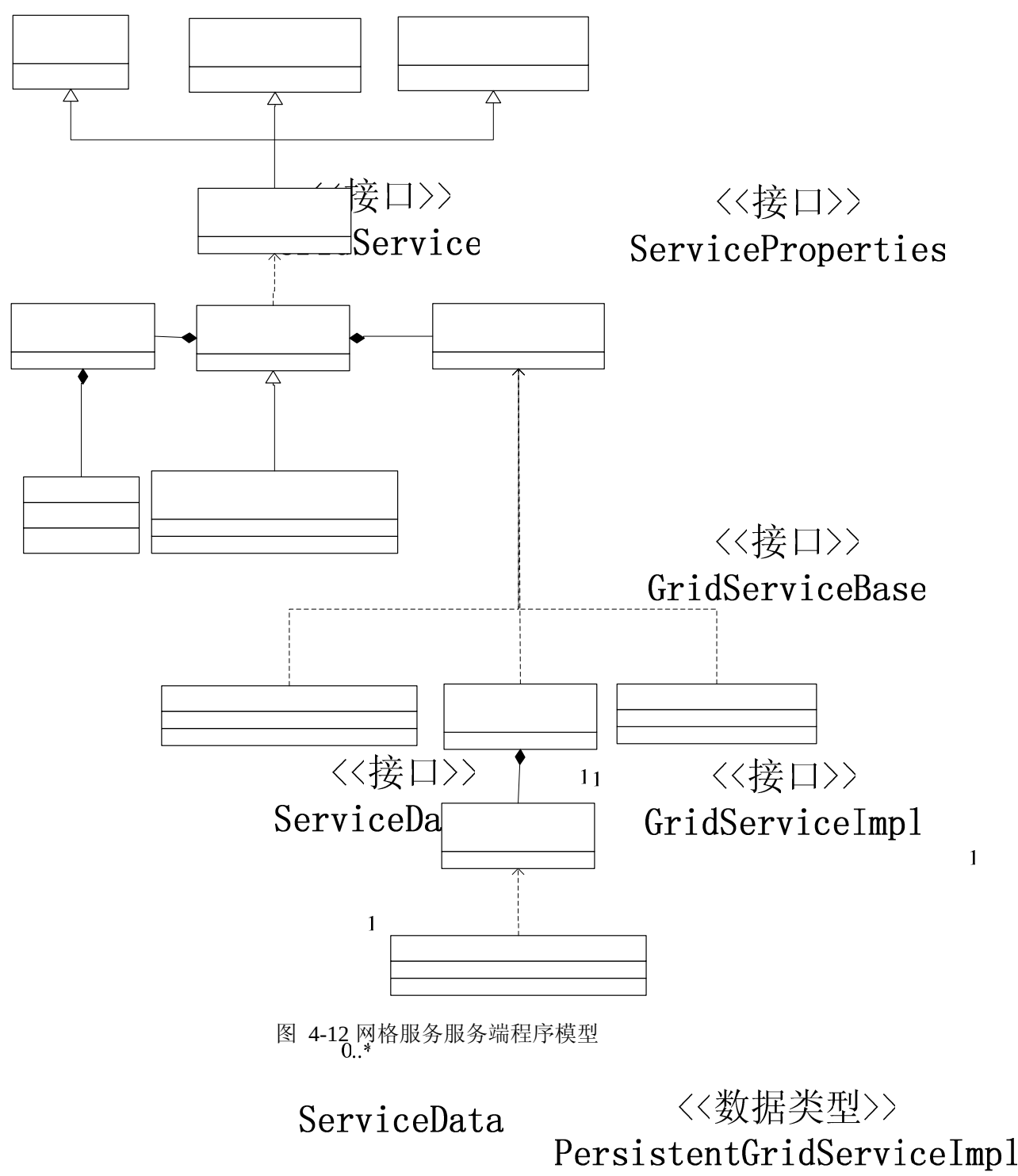


图 4-12 网格服务服务端程序模型

4. 4. Web 服务器端的 Apache 与 Tomcat

4. 4. 1. Web 应用程序目录结构及错误处理

Java Servlet 规范定义了目录的机构化分层体系，这些目录被用于部署时的打包文件。

每一个应用程序都有一个根目录，称之为环境路径，当部署不同的 Servlet 的时需要输入相应的环境路径。任何两个应用程序的环境路径都不能相同，因为这样会造成潜在的 URL 冲突。

在文档根目录下有一个特殊的目录叫做WEB-INF，WEB-INF目录中包含的内容如下^[11]：

Web.xml 文件 Web 应用程序部署描述符；

/classes 目录 用来存储 Servlet 和有用的类的类文件；

/lib 目录 Java 归档(JAR)文件的存储区域。

处理错误

错误可以分为两类：HTTP 错误和 Servlet 异常

HTTP 错误处理方法：

为应用程序编写错误页面来代替标准的 HTTP “404 Not found”错误页面。

发送重定向。如果出错的话，另一个 Servlet 可以做的有用的事是将客户重定向到另一个 URL。使用 HttpServletResponse.SendRedirect(string location)方法可以将响应重定向到指定的位置。

Servlet 异常处理方法：使用 try ...catch ...代码。

1. 给用户返回相应的错误信息

例如：

```
try{...}
catch(RemoteException e)
{
    res.sendError(res.SC_SERVICE_UNAVAILABLE, "internal communication error");
}
```

2. 将用户重定向到其他的 URL

例如：

```
try{...}
catch(RemoteException e)
{
    res.sendRedirect("/Servlets/ReportErrorPage")
}
```

4.4.2. Apache 与 Tomcat 连接

我们在一个 Web 服务器上使用 Java.Servlet 来处理 HTTP 请求，这些 Servlet 将会产生在浏览器上显示的 HTTP 响应。

Servlet 是一个服务器端组件，它可以用来扩展任何运行 Java 的服务器的功能，但是大多数 Servlet 是用来在 Web 服务器上编写 Web 应用程序的。Servlet 通常用来创建内容是非静态的页面，内容会根据用户输入或其他变量数据而改变。Apache 在处理静态页面方面占有优势。

Servlet 环境

Web 用户(Web 浏览器)使用 HTTP 请求/响应协议与服务器进行交互。Servlet 容器提供以下的服务和功能：

1. 用来发送请求和响应的网络服务；
2. 为 Servlet 注册一个或多个 URL；
3. 管理 Servlet 的生命周期；
4. 解码基于 MIME 的请求；
5. 构造基于 MIME 的请求；
6. 支持 HTTP 协议和其他协议。

Apache

Apache 在处理静态页面方面很有优势。

Jakarta提供了专门的组件可以将Apache和Tomcat连接起来：JK和JK2。具体实现根据要求安装配置就可以了^[12]。

Servlet 链和过滤器

根据具体的需要，在本 Web 服务器的实现过程中，用户在访问某些页面时，设计了 Servlet 链，相当于管道的方式，在服务器端实现了数据的接收然后解析转发给最终用户的过程。用 Servlet 还可以实现过滤器的功能，审计和更改浏览器请求与响应的格式。由于是 Applet 的直接访问，所以在具体的实现中没有涉及到过滤器的功能。Servlet 链的实现有三种方法：

1. 通过 URL 中用逗号分隔的多个 Servlet 来直接触发。实现上直接在 URL 中触发 Servlet 链的方法很简单，只要在 URL 后面将链中的 Servlet 用逗号按顺序隔开即可。
2. 多个 Servlet 指定同一个别名，然后通过别名方式来触发 Servlet 链。方法使用别名方式指定 Servlet 链，如采用 JRUN Servlet 引擎，则在 JRUN 的 Servlet URL Mapping 功能中进行配置。
3. 指定一种 MIME 类型，规定凡是这种 MIME 类型的响应都要通过某一个 Servlet 来过滤，这也是一. 种 Servlet 链的应用。

本设计采用了最简单的第一种方法。

Tomcat 处理多用户的选择

Servlet 需要处理许多个用户的请求。针对要不断查询数据的多用户请求的处理也十分关

键。

在只是简单应用的用户计数器上，用 Java Beans 来实现的。对于需要访问数据的操作，由于数据的访问现在只能有一个 Servlet 来访问这些资源，所以为了避免并发性事件，Servlet 使用了单线程的模式来是实现。

Servlet API 指定一个容器必须保证不存在两个实现了 `javax.servlet.SingleThreadModel` 接口的 Servlet 线程并发地运行。这意味着这个容器必须维护一个 Servlet 实例池并且将每个新请求分派给严格自由的 Servlet，一次只有一个请求被处理。

所以在实现的过程中只需要在类的申明处增加了 `SingleThreadModel` 接口就实现了我们的要求。

```
public class FitsServ extends HttpServlet implements SingleThreadModel{}
```

4.4.3. 用户 session 及图像转换

在 Web 服务器端，Tomcat、Apache 的配置上应用了前两小节的具体方案；除此之外，还实现了两个功能：

1. 用户 session。为了记录用户数，对 session 的支持，特用类 `UsrID` 来实现。

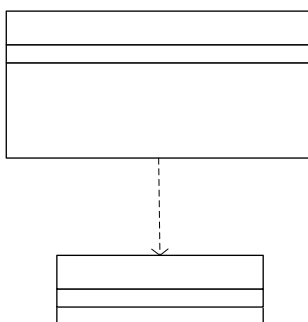


图 4-13 用户 Session 处理

这个 Servlet 首先访问一个记录了已经有多少个用户访问过此页面的文件，调用一些 Tomcat 配置中的常用属性值。在输出的页面，对某些图片进行编码处理，再将静态图像传给浏览器。

2. 对图像的处理。在 Java 中，`com.sun.image.codec.jpeg` 包实现了图像数据流和 jpeg 图像格式之间的转换。在网页的显示中，用到的较为频繁。

```
ServletOutputStream out = response.getOutputStream();
JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);
encoder.encode(image);
```

4.5. 客户端与服务器端(Applet 与 Servlet)通信方式对比选择

Applet 与 Servlet 之间的联系一般是间接的，即页面请求由浏览器发送给 Servlet，作为回

答, Servlet 将结果生成 HTML 文档发送给浏览器。

但如果要显示一些实时的动态数据, 如需求中的实时查询天文数据等, 建立两者之间的直接联系就很有用了。在客户端 Java 程序与服务器端 Java 程序之间建立这种复杂的交互方式仅仅使用 Servlet 与 Applet 提供的 API 是不够的。

下面的四种通讯方式中的任意一种都可以协助实现 Servlet 与 Applet 之间的通信: 通过 HTML 页面传递参数, 用 java.net 包的网路功能建立直接网络连接, 远程方法调用(RMI)与 CORBA。

4.5.1. 通过 HTML 页面传递 Applet 参数

通过 HTML 页面传递参数是在 Servlet 与 Applet 之间建立联系最为常用的方法。Servlet 只要把传递给 Applet 的参数写入 HTML 页面就可以了。这种通讯是单向的, 用于实现 Servlet 对 Applet 的控制, 用 Servlet 中的 doGet 方法即可输出 HTML 页面来实现某些功能。

这种方法很简单, 但它有两个缺点。首先, 所有参数都是静态的, 即一旦 HTML 页面发送完成, 要更新其中部分数据就很困难了。虽然可以使用 HTTP 的刷新机制, 但这种刷新需要重新下载整个页面, 效率是很低的。第二, 如果需要传递给 Applet 的数据量很大或数据结构很复杂, 就会使得 HTML 页面过于庞大或复杂了。

4.5.2. 用 java.net 包建立双向通讯

第二种方法是利用 java.net 包提供的网路能力。以连接流通讯方式为例, 服务器端的操作一般为:

1. 创建一个 ServerSocket 对象, 在指定端口监听客户端发来的请求;
2. 在接收到请求时 accept()方法将返回一个 Socket 对象;
3. 用上述 Socket 对象创建输入、输出流对象;
4. 通过输入、输出流与客户交互;
5. 交互完毕, 关闭输入、输出流与 Socket;
6. 服务程序运行结束, 关闭 ServerSocket。

在客户端的操作为:

1. 创建 Socket 对象建立与服务器的连接;
2. 用该 Socket 对象创建输入、输出流;
3. 与服务器交互;
4. 交互完毕, 关闭输入、输出流与 Socket。

这种方法只依赖于标准的 Java 网路支持, 不需要用到附加的软件包或工具, 因此显得相当简洁和灵活, 易于实现某些特殊的需要。但在另一方面, 服务器端对进入请求以及客户端对返回结果的分析工作量较大。如果服务器端要执行大量的操作, 可以考虑使用 RMI。

4.5.3. 远程方法调用 RMI

Java 的 RMI 技术显著地改善了客户程序操作复杂的服务器端对象的能力。对 Applet 来说，服务器端对象所提供的远程方法可以象普通的客户端句柄一样调用。

使用 RMI 时应先定义远程接口，即一个继承自 `java.rmi.Remote` 的公用接口。在该接口中定义要求在服务器上运行的方法，所有这些方法的 `throws` 子句至少应该声明 `java.rmi.RemoteException` 异常。

接下来是实现远程接口，远程接口的实现类应该：

1. 指定所实现的一个或多个接口。
2. 定义远程对象的构造方法。
3. 提供所有可供远程调用的方法的实现。
4. 创建和安装安全管理器。
5. 创建一个或多个远程对象实例。
6. 注册至少一个远程对象。

需要用到的包有 `java.rmi`，`java.rmi.server`。

除了编译 Java 代码之外，还要用 `rmic` 编译器生成远程对象的存根和骨架。存根(Stub)是远程对象在客户端的代理，它将 RMI 调用传递给服务器端的骨架(Skeleton)，后者负责将该调用传递给实际的远程方法。

例如执行命令 `rmic com.servletsolutions. RMIImplementation`，则生成的两个类文件分别为 `RMIImplementation_Stub.class` 和 `RMIImplementation_Skeleton.class`，前者封装入 Applet 且运行于客户端，后者封装入 Servlet 且运行与服务器端。

当大量的数据是由服务器端的对象动态提供时使用 RMI 是非常合适的。虽然 RMI 支持面向对象的分布式编程，但它要求客户端和服务端都是 Java 程序。

对系统的分析中得出，开发用到的是 Java，不需要使用混合语言开发的应用，所以 CORBA 在此例中不作具体的分析和比较。

4.5.4. 选择和实现

在对 Globus Toolkit 的分析中，可以发现，由于网格的服务器端有很多的东西需要处理。所以 Globus Toolkit 在处理客户端时的通信时常用的就是第三种方法 `rmi` 的方式。

对于 Applet 与 Servlet 之间的数据传输，HTML 页面也不是合适，所以在通信的选择上，用到了灵活的 `java.net` 包即建立了双工通信。

在 JSky 的原应用程序中，为了处理图像的方便，它在专门的目录(环境变量 `user_home`)下面建立了一个临时目录(`.jsky`)。将用户对下载图像和操作都记录在临时文件中。这样对图像的处理速度上就快了很多，并且用户在进行前进、后退等操作时，能够很快地恢复。

对于 Web 程序，Applet 要实现上述功能有两种方式：

1. Applet 数字签名。经过签名后的 Applet 可以使用本地资源，但用户为了系统安全，不信赖 Web 服务器网站颁发数字签名，那么 Applet 程序不能将访问过的图像存在本地，要进行快速访问已经访问过的数据的功能不能得以实现。
2. Web 服务器端提供一个临时的目录，每个用户都可以访问这个临时目录中的文件。再建立临时目录中的文件和某些操作图像的关联性。

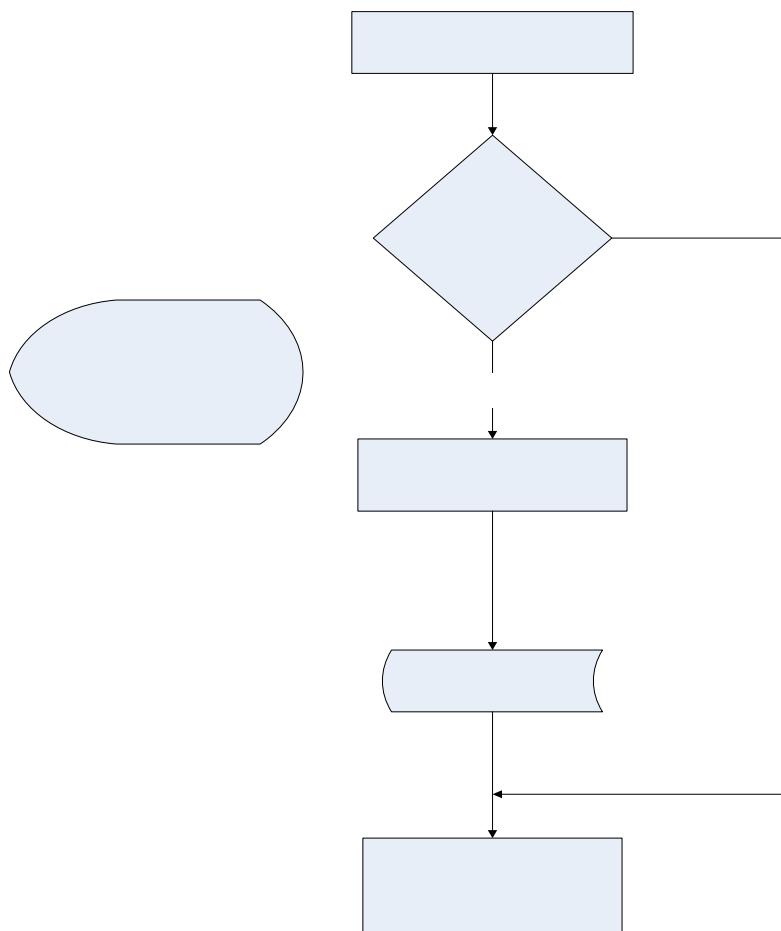


图 4-14 Applet 访问 Servlet 数据流程

图 4-14 说明：

1. 用户根据用户提交的请求访问 Servlet 页面，Servlet 根据传送过来的数据参数经度、纬度、星表名生成文件名名称，以字符串的方式表示。
2. 若图文件或者表文件不存在，则调用数据服务客户端，将从数据服务得到的数据按上述方式命名，并存储。若图文件或者表文件存在，不作任何处理。
3. 将文件数据通过 `java.net` 包写入客户端，`InputPanel` 类中的 `search()` 方法进行接收。

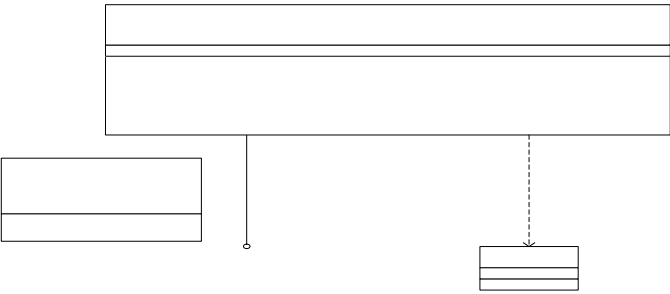
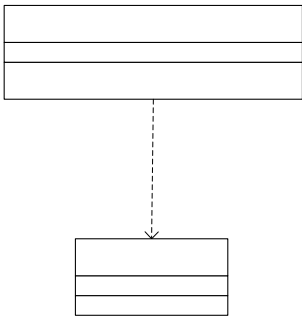


图 4-15 InputPanel 类图

InputPanel 继承了 JPanel 来接收用户的输入参数。对输入参数的限制由另外一个类 InputRestrict 来实现。



```
+InputPanel(in
+ActionPerform
+search() : vo
+testHMS() : v
```

<<接口>>
InputInterface
+setPara()

图 4-16 类 InputRestrict

InputRestrict 类通过 createDefaultModel 方法限制了用户输入的参数个数和参数范围，能够对用户的输入加以限制。

4. 6. VOTable 与 XML 的 Java 解析

4.6.1. FITS 说明及显示说明

FITS(Flexible Image Transport System)是目前天文学上标准的数据格式，它描述了数据定义和数据编码的一般方法，被用于多为矩阵数据，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。

每个 FITS 文件开头用了许多包含标题数据的逻辑纪录，每个逻辑纪录长度为 2880bytes/8bit。每个标题数据用 7 位 ASCII(8 位字节右对齐，每一字符的高位为 0)的 80 字符形式书写。

下面列出 FITS 头中将在处理过程中遇到的重要的几项：

#create

SIMPLE=T, 表明数据数组是用 8 位无符号, 或 16 位或 32 位二进制补码, 二进制整数用典型的字符序列写的。

NAXIS1=530

NAXIS2=530

CTYPE_n 表示第 n 轴物理坐标类型.....

CRVAL_n 表示某一个参考象元在第 n 轴上的物理坐标值

CRPIX_n 表示某一个参考象元在数据阵中第 n 轴上的象元位置(CRPIX_n), 象元位置在基于通常的 FORTRAN 意义上是在 1 到 NAXIS_n 之间, 但 CRPIX_n 是个浮点数, 因此参考象元位置可能并不对应一个实际的象元, 如 129.5 或 -3 的值也是可以接受的。

EPOCH(浮点数): 坐标系统的历元(以年为单位)。

BITPIX: 每个像素值的位数。

EQUINOX: 历元 B1950—FK4, J2000—FK5

在实现上, 已经有专门的 Java 包 nom.tam.fits 来读取 FITS 图。

4.6.2. VOTable 格式数据

VOTable 是一种标准的定义好的 XML 格式。它具有完整的结构, 由它的 Schema^[13]进行了描述。这个 Schema 实质上代表了 VOTable 的定义, 说明 VOTable 的文档必须毫无错误地通过 W3C XML Schema 的验证器。因此, XML 允许应用容易地使输入的文档有效, 通过 XSLT(eXtensible Style Language Transformation)引擎进行变换。

天文学家为了支持虚拟天文台在线数据的互操作性和可扩展性^[14], VOTable 为表格数据特别是天文表格数据设计成灵活存储和交换的格式。

虚拟天文台要用 VOTable 数据格式的原因主要是:

VOTable 有天生为了大的数据量和网格计算的特征。

数据存储灵活高效, VOTable 将数据的特征说明和数据部分分开了。这样数据部分的存储就很灵活, 存储的方式也多样。

VOTable 中的数据部分, 能够用三种不同的格式来表述: TableData、FITS、二进制。Tabledata 是纯粹的 XML 格式因此小的表能够容易地被 XML 工具处理

FITS 与 VOTable 格式比较

FITS 二进制表格是对天文学家来说耳熟能详, VOTable 能够用来或者包装这样一个文件, 或者重新打包(encode)元数据; 但对流 FITS, VOTable 包装很困难, 因为在 FITS 头中(NAXIS2 关键字)需要数据集大小, FITS 需要一个规范在可变长数列最大大小的前面。

BINARY 格式支持效率和使编程灵活, 不需要 FITS 库, 支持流的范例。

1. FITS 能实现但 VOTable 不能实现的

FITS 有复杂的语义, 能够将一个子串结构化, 变成为子串的集合, 但是 VOTable1.1 不支

持类似的功能。

现在版本的 VOTable 允许固定长度和变长的字串，也支持固定长度串的变长的排列。

2. VOTable 能实现但 FITS 不能实现的

VOTable 支持元数据(metadata)和数据的分离存储，以及其它分布式计算中的一些概念。它尝试正式地表达参数或域的语义内容。它有 XML 的层次性和灵活性：用 1.1 版本中介绍的 GROUP 元素，VOTable 中的列能够组在任意复杂的层次中，ID 属性能够用在 XML 中来作为关键内容的指针。

FITS 不能处理 Unicode 字符。

特殊说明的一点：

FITS 和 VOTable 数据间的转换是可逆的：任何 FITS 表能够不损失任何信息转换成 VOTable，得到的 VOTable 能够不损失任何信息转换成 FITS 格式。然而，建立新的不能转换成 FITS 格式 VOTable 不损失任何信息也是可以的。

4.6.3. 支持 Document Object Model (DOM) 的解析器

在处理 DOM 的时候，我们需要读入整个的 XML 文档，然后在内存中创建 DOM 树，生成 DOM 树上的每个节点对象。当文档比较小的时候，这不会造成什么问题，但是一旦文档大起来，处理 DOM 就会变得相当费时费力。特别是其对于内存的需求，也将是成倍的增长，以至于在某些应用中使用 DOM 是一件很不划算的事(比如在 Applet 中)。

4.6.4. 支持 Simple API for XML (SAX) 的解析器

SAX 是 Java 自然的、分析器无关、基于事件的 API，适合超大文档和流式数据。

当 SAX 解析器来解析 XML 文档时，解析器在文档的不同处将产生事件，这些地方包括：在文档开始和结束时，在一个元素开始和结束时，或者它在一个元素中找到字符时，以及其它若干点。在这些地方可以编写 Java 代码来处理每个事件，以及如何处理从解析器获得的信息。

SAX 接口中两个关键的接口是 XMLReader 和 ContentHandler。XMLReader 是分析器接口，ContentHandler 让分析器与客户应用程序通信。

所谓事件驱动，是指一种基于回调(callback)机制的程序运行方法^[16]。

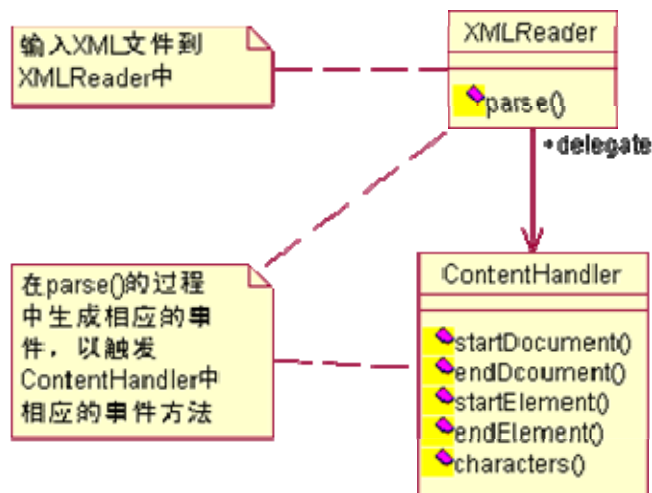


图 4-17 SAX 基于回调的解析机制

在 XMLReader 接受 XML 文档，在读入 XML 文档的过程中就进行解析，也就是说读入文档的过程和解析的过程是同时进行的，这和 DOM 区别很大。

解析开始之前，客户应用程序通过 setContentHandler() 方法向 XMLReader 注册 ContentHandler 接口的特定实例，也就是相当于一个事件监听器。

在后面当 XMLReader 分析文档读到合适的内容，就会抛出相应的事件，并把这个事件的处理权代理给 ContentHandler，ContentHandler 调用其相应的方法进行响应。

4.6.5. VOTable 解析在 Web 服务器端的实现

在解析 VOTable 进行的对比选择^[17]，如表格 4-1。

在实际的应用中，天文上已经有针对 VOTable 数据的解析包，JAVOT^[18] 和 SAVOT^[19]。JAVOT 是基于 DOM 机制开发的，在进行 Web 服务器上 VOTable 转换为网页界面时用到，在重新访问某些节点时是十分方便。但是速度很慢，当有五个用户同时访问这个页面的时候，服务器的反应时间就很长，用户难以接受。

另外一个包 SAVOT 是基于流的解析方式变换的。经过同样的数据测试，在同时有一二十用户访问页面的时候，解析速度要比 JAVOT 的速度快上十倍。因此在服务器提供多用户的访问过程中，SAVOT 基于 SAX 的解析无疑是最佳的选择。

我们在选择的时候，用到了 SAVOT 中的某些类，同时基于我们需要的变换进行了某些事件的处理，如图 4-18。

比较因素	DOM	SAX	VOTable 解析
是否在一台内存很少的机器上，工作	DOM 树耗内存较多	基于流的方式最佳选择	在 Applet 下面工作，内存少
是否需要 XML 源文件的少量内容	要为源文件中的每个东西创建对象	检查每个事件，是否需要相关找到需要的东西，代码可以抛出异常来完全停止解析	属性项很多，有的高达 85 项() 1. 在作图时只需要其中几项，SAX 合理 2. 显示时要从另外角度考虑
一旦解析了 XML 文档，是否需要多次访问这些数据	DOM 自动保存所有数据，用 DOM 比较合理	事件触发后，开发人员决定需要保存多少；若需要访问不曾保存的事件，需要再次解析	DOM 比较合理
应用程序如何部署	DOM 解析器比较大	SAX 比较小，适合 Applet 部署	Applet 程序要下载解析器，SAX 合适

表格 4-1 VOTable 解析选择

XMLConsumer 是 cds.xml 包中的一个接口，cds.xml 包是专门针对 VOTable 提供了一类 API。在解析 VOTable 数据的过程中，主要实现了 SAX 解析中提到的 startElement,endElement, characters 方法。在此过程中，设置了最小值 min 和最大值 max，主要是记录了 VOTable 属性值的个数，以确定界限。



图 4-18 VOTable 解析处理

4. 7. 数据图形

4. 7. 1. 世界坐标系及实现

理想情况下，世界坐标系应该和 FITS 文件格式无关。在 Java 中，这可以通过提供类构造器来实现，通过构造器提供必须的关键字值作为参数。

另外替代的方法是，通过哈希表包括关键字/值，传递图像属性对象给类构造器。世界坐标系中所有重要的映射应该被支持。

关于 FITS 图像，一些重要的映射都被包含在其中了，这些映射的方法也可以在 JAI 的映射中用到。

一个类应该基于图像头中的值，提供方法来转换图像和坐标系统，。

一个类应该代表一个给定历元里的(ra, dec)数值，提供方法来读写各种格式中的世界坐标系。

这些需求在一些开发包中已经被实现了。具体可以引用其中的一些包。现在我们知道的两种 Java WCS 实现在这里可能会有用。

FITSWCS包，是基于Calabretta的C语言写的wcslib包开发出来的，它已经有Java版本的^[17]。

Gemini 的 Observing Tool 包括较早 Doug Mink 的 wcssubs 包。

另外一个替代的可能用最新的 wcssubs 包作为本地代码，至少到现在一个 Java 版本是可以得到

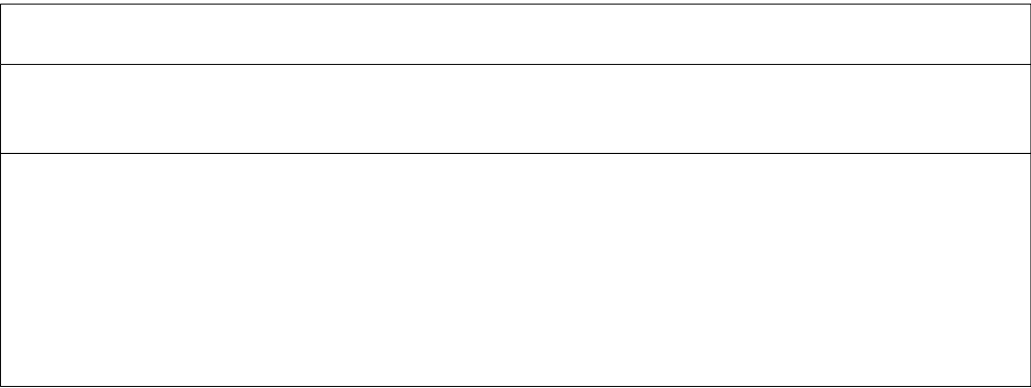


图 4-19 用户输入坐标和世界坐标系

在此坐标系中，引用了 jsky.coords 包中的 WCSTransform 来实现了三个参数到世界坐标系的变换。

4.7.2. FITS 对 JAI 支持实现

JAI 对 FITS 的支持在 JSky 的包中即包含了如此的功能。

显示 FITS 图像操作，一般的读取操作和对 JAI 的支持，需要进行的改变。

一般 FITS 图的读取显示要求	JAI 支持 FITS 的变化
图像数据返回为值的二维数组	需要将图像数据作为一维数据值来访问
图像数据被读成一个新的数组	应该能够提供一个数组，数组值为图像数据
整个图像一次被读到	能够仅仅读取被选中区域的图像数据，对 JAI 提供支持

表格 4-2 JAI 对 FITS 支持要求的变换点

第一步：读取 FITS 文件，现在只是作为数组能够将 FITS 图中的数据取出来，还不能看到图像。

第二步：依照像素比例处理图片，使其可见。

JAI 支持图像处理链的概念，即管道。在这个地方，用户可以先进行或申请许多项操作，最后得到显示的图像。

在 JSky 中，这样的方法被处理为一个类 ImageProcessor。

一个 ImageProcessor 对象有一个源图像、一个显示图像、以及许多图像转换方法。这样通过一系列的变换，源图像就转换成了显示图像。

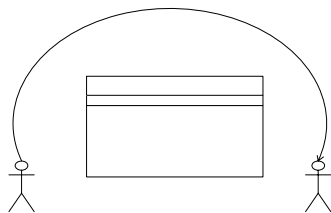


图 4-20 从源图像到显示图像的处理

图 4-20 就是经过了轴翻转、旋转、颜色变换等变换，将源图像进行了转换。

对图像进缩放操作是由显示图像的类来完成的，因为一张图可能同一时间在不同的分辨率在不同的窗口中显示。

JAI 或者用户定义的图像处理操作可以在显示图像前增加到图像处理链中。

4. 7. 3. 用户界面

图像显示的用户接口

JAI 包中没有合适的显示小窗口，里面有一些简单的示例小窗口，但是包中主要的仍然是

关注图像处理而不是显示。尽管基于小窗口的 AWT 中，比如说画布在图像显示方面有很高的效率，但经验显示，在显示图像的时候，我们不得不需要通过 Swing 小窗口提供的双缓冲 (double buffering) 技术来避免图像的闪烁，特别是在图像上拖动图形图标时。

现在的原型是 ImageDisplay 类，它基于 Swing JPanel 类显示图像。这个类用来显示主要图像，还有面板和缩放窗口。

侦听变换事件

因为任意个窗口可以在某个给定的时间显示图像或者做图像统计，所以必须有一种方式来通知这些窗口这副图像的变化。由于这个原因，ImageProcessor 类允许你登记一个 ChangeListener，这样在图像改变的时候能够被通知到。这种机制用来保持面板小窗口中显示图像与主要窗口中显示图像的同步。

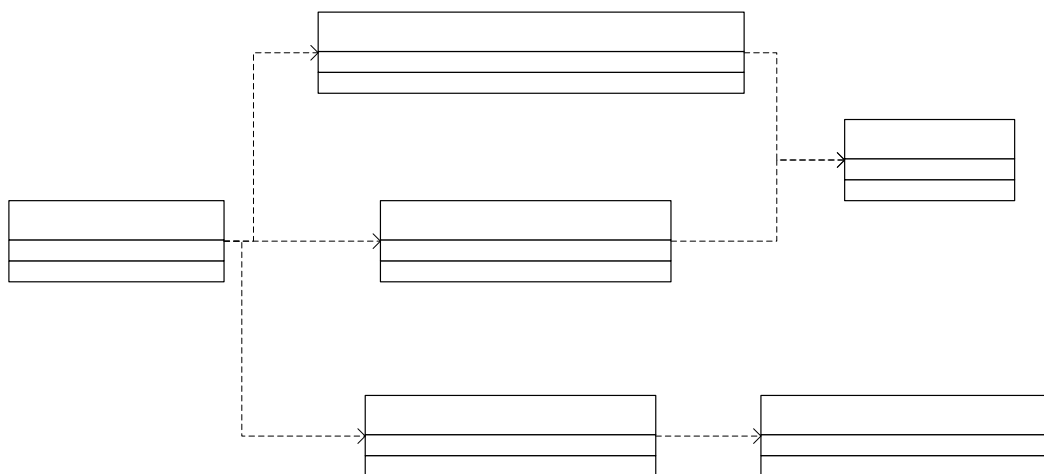


图 4-21 用户界面功能类图

用户界面的主要由几个面板组成，输入面板、画图画布、图像处理面板。在用户输入参数出错时，会抛出出错会话窗口。画图程序和图像的处理一直依赖坐标系统来完成具体的处理过程。

ImageDisplay

4.8. ConeSearch 例子

前面几节中已经对 ConeSearch 的例子做了详细分析，以及对关键技术及实现做了详尽的分析。ConeSearch 的例子是中国虚拟天文台做的最初的一个尝试，在这个过程中，我们取得了一定的进展，主要是在调通实例上。随着 Globus Toolkit 的发展，现在我们在注册和发现方面也跟随着有了一定的发展。

MySky (Applet)

StarCanvas

ConeSearch 的可视化实现的例子工作的例图如图 4-22:

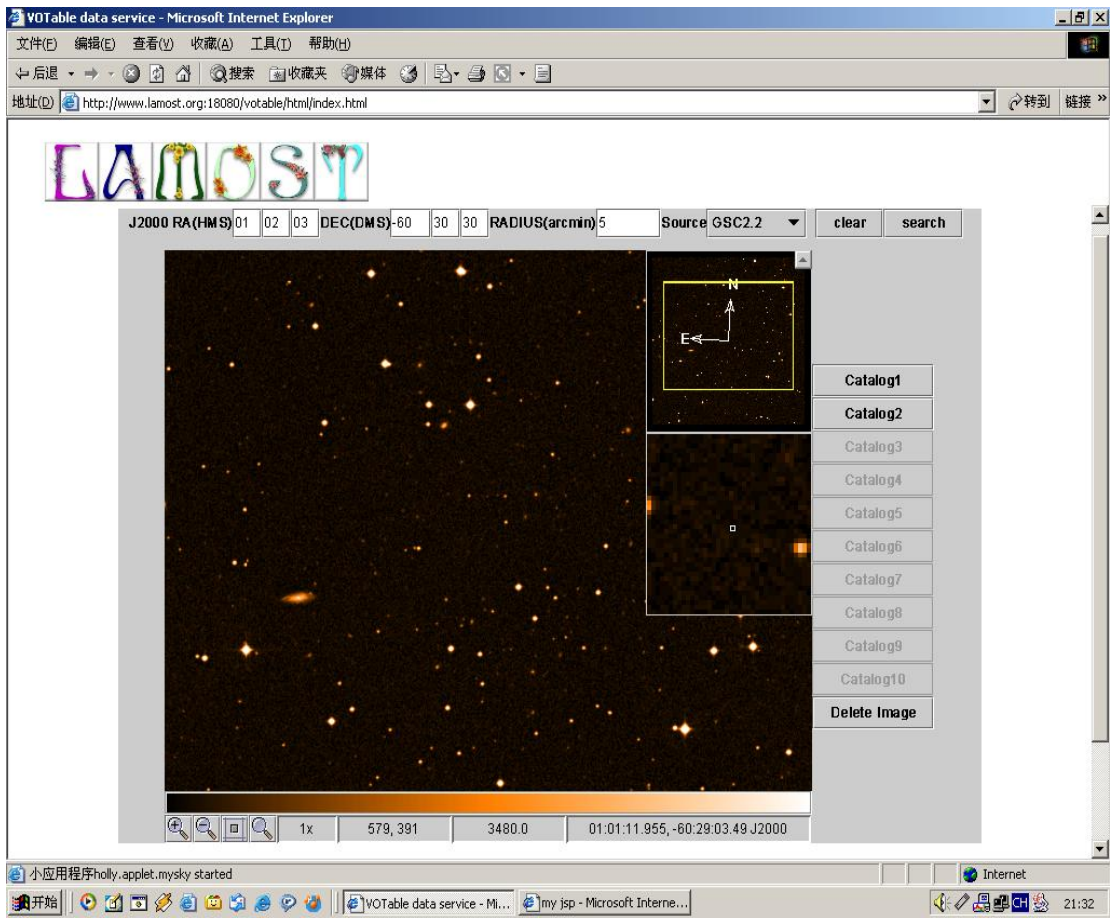


图 4-22 ConeSearch 主界面图

my.jsp - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) http://211.167.66.198:18080/votable/jsp/Gsc.jsp?xmlPath=GSC2.20010342029491700833.xml&Ra=1.0342&Dec=-60.5083&Radius=0.0833&Catname=GSC2.2

数据表

ID_MAIN	POS_EQ_RA_MAIN	POS_EQ_DEC_MAIN	POS_ERROR	PHOT_PHG_MAG	MAG_ERROR	NOTE	CLASS_OBJECT	ID_PLATE	FLAG
S01212135954	000.943398	-60.450520	1991.852	0.309	0.299	16.33	0.40	17.14	0.27
S01212135870	001.126593	-60.459372	1991.852	0.309	0.299	16.77	0.40	17.53	0.27
S01212135728	001.104467	-60.476047	1991.852	0.309	0.299	14.68	0.40	15.64	0.27
S01212135707	001.022626	-60.477251	1975.836	0.273	0.291	19.44	0.28	0	1.39
S01212135675	001.073619	-60.480579	1991.852	0.309	0.299	18.06	0.40	19.49	0.28
S01212135664	001.021104	-60.482281	1991.852	0.309	0.299	16.15	0.40	17.95	0.27
S01212135532	001.128112	-60.494807	1975.836	0.273	0.291	18.90	0.28	3	1.77
S01212135493	001.115005	-60.498702	1991.852	0.309	0.299	15.52	0.40	16.27	0.27
S01212135326	001.126071	-60.517106	1991.852	0.309	0.299	17.34	0.41	18.43	0.28
S01212135308	001.122060	-60.518820	1991.852	0.309	0.299	17.89	0.41	18.79	0.28
S01212135179	001.127451	-60.531350	1975.836	0.273	0.291	19.26	0.29	3	1.83
S01212135351	001.076567	-60.515479	1991.852	0.309	0.299	14.85	0.40	15.84	0.27
S012121367	001.052903	-60.487642	1991.852	0.309	0.299	13.09	0.40	13.97	0.27
S01212135989	001.126299	-60.446802	1991.852	0.309	0.299	16.92	0.40	17.69	0.27
S01212135642	001.157391	-60.484739	1991.852	0.309	0.299	17.13	0.40	17.68	0.27
S01212135586	001.193450	-60.489052	1975.836	0.273	0.291	19.30	0.28	3	1.97
S012121370	001.194276	-60.495777	1988.820	0.066	0.062	12.43	0.15	11.90	0.13
S012121376	001.183069	-60.510695	1991.852	0.309	0.299	13.08	0.41	13.89	0.27
S012121373	001.153831	-60.504289	1991.852	0.309	0.299	12.58	0.40	13.50	0.27
S01212135245	001.193513	-60.525717	1991.852	0.309	0.299	16.17	0.41	17.42	0.27

图 4-23 ConeSearch 数据表图

http://www.lamost.org:18080/votable/xml/GSC2.20000000090000000833.xml - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) http://www.lamost.org:18080/votable/xml/GSC2.20000000090000000833.xml

```

<?xml version="1.0" ?>
<!DOCTYPE VOTABLE (View Source for full doctype...)>
<VOTABLE version="1.0">
  <DESCRIPTION>This files generate by the China Virtual Observatory!(http://www.china-vo.org) In case of any problem,please
  contact the author: Sang Jian,please Email to sang@lamost.org</DESCRIPTION>
  <DEFINITIONS>
    <COOSYS ID="Coosys" equinox="J2000" system="ICRS" />
  </DEFINITIONS>
  <RESOURCE name="GSC2.2" type="results">
    <DESCRIPTION>The Guide Star Catalogue, Version 2.2.01,1/271, The GSC II is an all-sky catalog based on 1 resolution scans
    of the photographic Sky Survey plates, at two epochs and three bandpasses, from the Palomar and UK Schmidt
    telescopes (DSS). Positions, magnitudes, and classifications are produced for all objects on each plate. The objects are
    then loaded into the COMPASS database, where multiple observations of the same object are matched and assigned a
    unique name. The GSC2.2 is a preliminary version of GSC II generated to support telescope operations at the GEMINI and
    VLT telescopes. The magnitude limits are 18.5 in photographic F (red) or 19.5 in photographic J (blue) or V. The
    parameters of the bright objects, overexposed on the Schmidt plates, are taken from the Tycho-2 (1/259) catalog. The
    final version (GSC 2.3), expected to be released in 2002, will also contain proper motions.</DESCRIPTION>
    <PARAM ID="CenterRA" unit="deg" datatype="double" name="CenterRA" value="0.0">
      <DESCRIPTION>The RA value of search center point!</DESCRIPTION>
    </PARAM>
    <PARAM ID="CenterDec" unit="deg" datatype="double" name="CenterDec" value="0.0">
      <DESCRIPTION>The Dec value of search center point!</DESCRIPTION>
    </PARAM>
    <PARAM ID="searchRadius" unit="arcsec" datatype="float" name="searchRadius" value="299.88">
      <DESCRIPTION>The search radius value!</DESCRIPTION>
    </PARAM>
    <LINK content-role="doc" title="documentation" href="http://cdsweb.u-strasbg.fr/viz-bin/Cat?1/271" />
  </RESOURCE>
  <TABLE>
    <FIELD ID="GSC22" datatype="char" name="GSC22" ucd="ID_MAIN" arraysize="14">
      <DESCRIPTION>Identification of the object</DESCRIPTION>
    </FIELD>
    <FIELD ID="RAdeg" unit="deg" datatype="double" precision="E6" width="10" name="RAdeg" ucd="POS_EQ_RA_MAIN">
      <DESCRIPTION>Right Ascension in ICRS (J2000), at Epoch</DESCRIPTION>
    </FIELD>
    <FIELD ID="DEdeg" unit="deg" datatype="double" precision="E6" width="10" name="DEdeg" ucd="POS_EQ_DEC_MAIN">
      <DESCRIPTION>Declination in ICRS (J2000)</DESCRIPTION>
    </FIELD>
  </TABLE>
</VOTABLE>

```

图 4-24 ConeSearch 中的 VOTable 数据

参 考 文 献

- [1] [JAI] Desktop Java Java Advanced Image(JAI) API. <http://java.sun.com/products/java-media/jai/>
- [2] The Tomcat 4 Servlet/JSP Container. <http://jakarta.apache.org/tomcat/>
- [3] Simon Brown 等著, 王军等译. JSP 编程指南. 第2版. 北京: 电子工业出版社, 2002
- [4] Eclipse-Overview. <http://www.eclipse.org/>
- [5] [ANT] <http://ant.apache.org/>
- [6] 柴晓路. Web 服务架构与开放互操作技术. 第1版. 北京: 清华大学出版社, 2002. 84-166.
- [7] [GWSDL] S.Tuecke, K.Czajkowski, et al. Open Grid Services Infrastructure. 2003-04-05.
- [8] [GGF] <http://www.ggf.org/>
- [9] The Globus Toolkit 3 Programmer's Tutorial. <http://www.casa-sotomayor.net/gt3-tutorial-working/>
- [10] Thomas Sandhom, Jarek Gawor. Globus Toolkit 3 Core- A Grid Service Container Framework:13-14
- [11] Martin Bond 等著, 张建明等译. 21 天学通 J2EE. 第1版. 北京: 人民邮电出版社, 2002. 372.
- [12] Tomcat-Apache using JK2 connector. http://www.greenfieldresearch.ca/technical/jk2_config.html
- [13] [VOTable-1.1.xsd] <http://vizier.u-strasbg.fr/xml/VOTable-1.1.xsd>
- [14] VOTable Formate Definition Version. <http://www.ivoa.net/Documents/PR/VOTable/VOTable-20040322.html#ToC2>
- [15] 扬帆. Java 与 XML 联合编程之 SAX 篇. http://www0.ccidnet.com/tech/guide/2001/10/08/58_3392.html, 2001-10-08
- [16] XML 认证教程, 第7部分: 解析器. <http://www-900.ibm.com/developerWorks/cn/xml/x-cert/part7/index.shtml>
- [17] Java, Image Browsing, and the NCSA Astronomy Digital Image Library <http://www.cv.nrao.edu/adass/adassVI/planter.html>
- [18] JAVOT:a Java Parser for VOTable. <http://www.us-vo.org/VOTable/JAVOT/>
- [19] VOTable/java/SAVOT. <http://nvo.gsfc.nasa.gov/viewcvs/viewcvs.cgi/VOTable/java/SAVOT/?cvsroot=NVO>

第5章. 总结和展望

本论文从虚拟天文台的产生和可视化的必要性说起，着重介绍了星表和图可视化的内容以及相关的技术和实现。本论文包括的内容有：

- 1.第一章介绍了虚拟天文台背景，可视化现状和在天文领域内各方面的应用。
- 2.第二章讲述了可视化星表和图的大概需求。
- 3.第三章对可视化作了详尽的设计，主要设计到用户界面的设计及需要达到什么样的要求。
- 4.第四章对功能的实现关键技术做了详尽的分析和介绍。这些主要的技术及实现的功能包括运用 Java Swing, AWT, JAI 实现用户界面，对 VOTable 格式数据的解析，对 FITS 图像的支持，Web 服务的实现，利用网格通知框架机制实现数据服务实例。最后介绍了可视化和数据网格的例子 ConeSearch。这是虚拟天文台初步探索的例子。

天文数据可视化包括的范围比较广，除了本文介绍的星表和图的内容以外，还有其他数据挖掘、光谱分析等等的可视化，这些也是虚拟天文台可视化要做的工作。

可视化不仅仅是指用户的界面的开发，更重要的是对各个领域的知识的熟悉和实现。比如文中提到的 ConeSearch 的可视化，除了对天文星图和星表要熟悉以外，还需要对坐标系统，图像的基本的颜色、直方图等图像处理的内容有一定的了解。当然，虚拟天文台作为分布式数据网格的应用，要在上面实现可视化，更要掌握与网络相关技术，如 XML, SOAP, Web Service, Apache, Tomcat 等，并应用这些技术实现某些功能。

可视化工具要发挥其重要的功能，提供给不同的用户使用，要针对专业用户、普通爱好者用户的需求进行分类和升级。可视化工具作为软件，需要遵循软件开发的规律和要求，有好的需求分析，尽量使用了模块化的设计方案，便于模块的优化和升级。

Globus Toolkit 平台的应用放宽了对开发语言的限制要求。现有的部分工具可以借助 Java 技术来发挥作用，而不是都需要重新设计和开发。

在可视化的需求调研过程中，JSky 和 Aladin 提供的功能给了我不少启发；在程序的设计和实现过程中，JSky 中的图像处理包经常被用到。但是某些 JSky 应用程序包要为 Web 程序所用，在处理方式上要一定的技巧来实现。例如，JSky 在实现过程中，使用了安装程序的本地资源或者某些文件中的属性；在 Web 程序中利用哈希表的实现了相应的功能。

可视化工具的开发，要达到能够完善、好用的目的，除了在设计 and 开发过程中尽量考虑到性能稳定、系统开销、程序的扩展性以外，更需要在应用中不断地测试并完善各方面的功能。

星表和星图的可视化目前已经实现了查询数据、图像可视化、对多用户支持等大部分基本的功能，图层的叠加以及对多星图的支持等功能还需要进一步的研究和开发实现，这也是将来虚拟天文台可视化的工作重点之一。

致 谢

转眼间就要毕业了，时间过的真快。想当初我研一的时候还在为自己的发展方向摇摆不定，在其中倍受煎熬。在这过程中，我要感谢汪景绣老师对我的宽容，赵刚老师对我的鼓励，彭勃老师对我选择方向提供的建议。

我最要感谢的是带领我做虚拟天文台方向的赵永恒老师。赵永恒老师把握方向的能力很强，对天文技术方面研究的很深入。开始在虚拟天文台实现的初步探索上，大家知道的都不是很多，资料也很少，做起来很费劲，但是在赵老师的带领下，能够一直努力自力更生，积极与其他兄弟单位合作，为我们的研究和开发工作带来了很大的信心。赵永恒老师广博的胸襟，为学生解答问题的严谨态度亦使我在研究过程中学到了不少知识。

在此我要感谢开始为中国虚拟天文台作了很长时间调研工作的崔辰州和张彦霞。崔师兄在遇到困难时的勇敢自信和张彦霞在科研中的刻苦敬业精神给我们树立了很好的榜样。

另外我要感谢最初与我进行 Demo 开发、经常在实验室干通宵的桑健同学，后来加入虚拟天文台开发工作经常进行技术探讨的王小倩、李长华、刘高潮；感谢在我研究生期间不厌其烦给我答疑解惑的张昊彤师兄，罗阿理、赵瑞珍和石火明副研究员。

感谢在网格开发上给我启发的清华大学刘鹏、中科院网络中心的南凯、罗泽、许海燕，冯家宏。

尽管在实验室还有一些人做的方向不尽相同，但我们平常之间的探讨也让我受益匪浅，在此我要感谢他们。他们是吴潮、王伟、卞维豪、覃冬梅、许馨、刘中田、朱光华、贾磊、李红星。

在研究生三年的生活当中，我要感谢国家天文台老师和工作人员的辛勤劳动，他们的工作作为学生的我创造了很多有利的条件。首先我要感谢杜红荣老师。我在研究生期间碰到了事情，她总是想方设法地为我着想尽力解决。我要感谢孙盛慈老师，尽管我还没有入党，但是在她的鼓励和影响之下，我的思想得到了很大的提高。感谢 LAMOST 项目的老师和工作人员苏洪均研究员、王钢研究员、陈英老师、袁晖老师、门力老师、李有刚老师、冯磊师傅。

感谢同级的白光顺、刘刚、刘颖、周玮、王杰、刘飞飞、胡新华、郭娟、贾志宏，大家共同度过了美好的三年。

感谢我大学老师韩俊，以及在韩老师实验室一同工作过同在北京发展的李卫国、王中勇、胡建军，好友李洁、刘怡芳、梁晏珍。

我要特别感谢李耀伟，在我研究生期间在技术上对我的指引和支持，在精神上对我的鼓励和生活上对我的关心。

衷心地感谢我的父母对我读研究生的支持，我会一直努力来回报你们。感谢我的老妹，我们俩经常互相鼓励，希望你有美好的前程。另外，我要感谢我的爷爷，他在我找工作比较关键的时候病危，卧病在床仍鼓励我好好努力，愿他在九泉之下安息！

发表论文

邵惠娟, 赵永恒. 中国虚拟天文台可视化服务. 天文研究与技术, 2004(2): 152-159