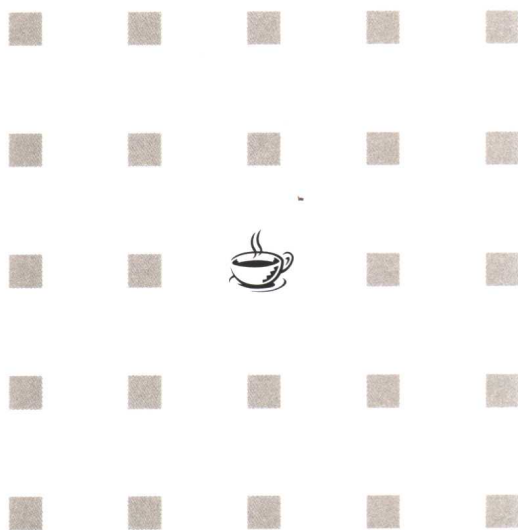


Advanced and Practical  
Java Programming

# Java

## 高级实用编程

管贻生 编著



清华大学出版社

# Java 高级实用编程

Advanced and Practical Java Programming

管贻生 编著

清华大学出版社

北京

## 内 容 简 介

本书以 Java 的实际工程应用为背景, 围绕作者在加拿大阿尔伯达大学计算机系开发的基于 Internet 的机器人多指手远程操作系统所涉及的技术, 对如何运用 Java 进行实际应用开发展开了深入全面的讲解。全书重点讨论五个问题, 分别是 Java 图形用户接口编程、Servlet 网络编程、Java 与数据库的连接、JNI 技术以及 Java 与 VRML 的连接。

本书内容充实、条理清晰, 是目前在 Java 编程的高级和实用方面不可多得的优秀教材。本书适合具有一定 Java 和 C/C++ 基础的读者阅读, 可作为从事 Java 实际编程应用的技术人员的参考书, 也可作为本、专科计算机相关专业及各类计算机学校学生在 Java 高级应用方面的教科书。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

### 图书在版编目(CIP)数据

Java 高级实用编程/管贻生编著. —北京: 清华大学出版社, 2004.1

ISBN 7-302-07687-1

I. J... II. 管... III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 106047 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-6277 0175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-6277 6969

组稿编辑: 丁 岭

文稿编辑: 孙建春

封面设计: 付剑飞

印 刷 者: 北京密云胶印厂

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 21.5 字数: 534 千字

版 次: 2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

书 号: ISBN 7-302-07687-1/TP·5632

印 数: 1~4000

定 价: 38.00 元

# 前 言

我们知道，Java 语言的历史很短，但它却以迅猛的发展速度成为计算机的主流编程语言。究其原因，除了其固有的一系列诸如面向对象、跨平台、鲁棒性、安全性、多线程性、自动收集垃圾以及便于网络编程等优点外，还在于它赶上了 IT（信息技术）时代的最好时机。这样 Java 不仅在计算机编程教育中被选为主要语言，而且在商业、通信和工业等很多领域中都开始或者已经得到了大量应用。可以说，掌握了 Java 就掌握了 IT 时代的强有力的工具。

目前大部分关于 Java 的书都只作一般性的普及介绍或者只就某个方面作专门讲述。而 Java 的实际应用往往涉及多个方面，所用的技术也绝不只是入门水平。如何综合运用 Java 的多项技术来进行实际的应用开发是每个 Java 开发人员都不能回避的问题，因而，本书讲解的 Java 高级实用编程会对 Java 开发人员有所裨益。

本书以实际的工程应用为背景，讲述 Java 高级的和较有实用价值的几个方面，包括 Java 的图形用户接口（GUI）编程、Java Servlet 网络编程、Java 与数据库的连接（JDBC）、Java 与 C/C++ 混合编程（Java Native Interface），以及 Java 与虚拟现实造型语言 VRML 的连接。主要内容围绕作者在加拿大阿尔伯达大学（University of Alberta）计算机系开发的基于 Internet 的机器人多指手远程操作系统 UA Telehand 所涉及的 Java 技术而展开，分 GUI、Servlet、JDBC、JNI 和 EAI 五个专题，每个专题自成一章，各章由 Telehand 系统互相串接和有机集成起来。作者参阅了大量的国内外有关资料，并总结了对 Java 在实际工程应用中的理解和体会。正如书名所示，本书不是一般的 Java 入门介绍书，读者应具备一定的 Java 和 C/C++ 的基础。相信本书对具备这一条件的读者会有所启示和帮助。

作者首先要衷心感谢加拿大阿尔伯达大学计算机系张宏（Hong Zhang）教授。张宏教授不仅为作者开发 Telehand 系统提供了平台和资助，还在系统开发的整个过程中给予了精心指导和高度关注。应该说，没有张宏教授的帮助和支持就不会有本书的写作和出版。

Teresa Hoe 在 Telehand 系统开发前期给予过技术支持。赵逸群女士在本书的修改和校阅过程中给予了大力帮助，提供了大量信息，提出了很多宝贵的建设性意见。熊妍妍、丁岭和朱润生诸君在本书的出版过程中也给予了大力帮助和关注。作者对在本书写作和出版过程中给予过帮助的各位朋友表示诚挚的感谢。

本书的构思得到了中国工程院院士、北京航空航天大学张启先教授的肯定和支持。不幸的是，张教授在本书开写之前因病辞世。作者谨以本书深切纪念张教授逝世一周年。

因作者的经验和水平和写作时间所限，书中难免有疏漏、不妥甚至错误之处，欢迎读者批评指正。

管贻生

2003 年 5 月

# 目 录

第 1 章 绪论 .....	1
1.1 Java 简介 .....	1
1.2 Java 高级应用举例：UA Telehand 系统.....	3
1.2.1 Telehand 系统的体系结构 .....	3
1.2.2 Telehand 系统的功能和操作 .....	5
1.2.3 Telehand 系统的实现和集成 .....	7
1.3 本书的内容和组织结构 .....	8
第 2 章 Java 图形用户接口编程 .....	10
2.1 引言 .....	10
2.2 Swing 总览 .....	11
2.3 顶级容器 .....	13
2.3.1 JFrame8 .....	13
2.3.2 JApplet.....	14
2.4 GUI 的总体效果 .....	16
2.4.1 基本的布局管理 .....	16
2.4.2 外观 (Look and Feel) .....	18
2.5 文本输入和显示 .....	20
2.5.1 文本域 (Text Field) .....	21
2.5.2 文本区 (Text Area) .....	25
2.5.3 样式文本区 (Styled Text Area) .....	26
2.6 按钮、复选框和单选按钮 .....	28
2.6.1 按钮 (Button) .....	28
2.6.2 复选框 (Check Box) .....	30
2.6.3 单选按钮 (Radio Button) .....	32
2.7 组合框 (ComboBox) .....	33
2.8 列表框 (List) .....	34
2.9 滚动条和滑动块 .....	39
2.9.1 滚动条 (Scroll Bar) .....	39
2.9.2 滑动块 (Slider) .....	41
2.10 标签 (Label) .....	42
2.11 边框 (Border) .....	44
2.12 任务进度的监视 .....	48

---

2.12.1	进度条 (Progress Bar)	48
2.12.2	进度监视器 (Progress Monitor)	49
2.13	表格 (Table)	50
2.13.1	一个简单表格	51
2.13.2	表格的行列操作	52
2.13.3	表格模型	55
2.13.4	单元的显示和编辑	57
2.14	树形结构 (Tree)	61
2.14.1	一个简单的树形结构	61
2.14.2	树的编辑	64
2.14.3	树节点的显示	66
2.14.4	树的事件监听和处理	68
2.14.5	树模型的定制	70
2.15	高级布局管理	71
2.15.1	网格布局 (Grid Layout)	72
2.15.2	方框布局 (Box Layout)	73
2.15.3	网格袋布局 (Grid Bag Layout)	75
2.15.4	卡片式布局 (Card Layout)	77
2.15.5	布局管理器的定制	79
2.15.6	布局管理器的选用	81
2.16	菜单 (Menu)	81
2.16.1	菜单的创建	82
2.16.2	菜单选项图标、键盘助记符和快捷键	84
2.16.3	菜单选项类型	85
2.16.4	菜单选项的有效性	86
2.16.5	弹出式菜单 (Pop-up Menu)	86
2.16.6	菜单布局的定制	87
2.17	对话框 (Dialog)	89
2.17.2	对话框的创建和用户数据输入	91
2.17.3	文件选择对话框	92
2.17.4	颜色选择器	96
2.18	控件组织器	98
2.18.1	分隔块 (Split Pane)	99
2.18.2	Tabbed Pane	100
2.18.3	工具条 (Tool Bar)	101
2.18.4	内部窗口 (Internal Frame)	103
2.19	Telehand 系统的 GUI 设计	105

---

第 3 章 Servlet 网络编程 .....	110
3.1 网络技术简介 .....	110
3.1.1 Web 技术的发展简史 .....	110
3.1.2 Servlet 的特点 .....	113
3.1.3 HTTP 基本概念 .....	113
3.2 开发环境的创建和使用 .....	116
3.2.1 Tomcat 的安装和运行 .....	116
3.2.2 Apache 的安装和运行 .....	117
3.2.3 Tomcat 与 Apache 的连接 .....	119
3.2.4 Tomcat 开发环境的使用 .....	122
3.3 HTTP Servlet 基础 .....	124
3.3.1 一个简单的例子 .....	124
3.3.2 Servlet API .....	126
3.3.3 Servlet 的运行机制 .....	128
3.3.4 Servlet 的生命周期 .....	128
3.3.5 表单数据和 POST 请求的处理 .....	130
3.4 信息的获取 .....	131
3.4.1 初始化参数的获取 .....	132
3.4.2 服务器信息的获取 .....	135
3.4.3 客户信息的获取 .....	138
3.4.4 请求信息的获取 .....	140
3.5 HTML 页面的传送 .....	143
3.5.1 响应的基本结构 .....	144
3.5.2 HTML 页面的生成 .....	144
3.5.3 状态码和 HTTP 头部信息 .....	146
3.6 Cookie 和 Session .....	150
3.6.1 Cookie .....	151
3.6.2 Session .....	155
3.7 图像的产生、处理和传送 .....	157
3.7.1 静态图像的产生和传输 .....	158
3.7.2 动态图像的产生和传输 .....	160
3.7.3 图像的处理和传输 .....	163
3.8 多种语言的处理 .....	167
3.8.1 中文的显示 .....	168
3.8.2 多种语言的混合 .....	169
3.9 网络的安全性 .....	171
3.9.1 服务器的 HTTP 认证 .....	172
3.9.2 定制的认证 .....	173
3.9.3 基于表单的认证 .....	174

3.9.4	SSL .....	177
3.10	Applet 与 Servlet 之间的通信 .....	178
3.11	Servlet 与 JSP .....	189
3.11.1	JSP 的特点和工作机制 .....	189
3.11.2	一个 JSP 的简例 .....	191
3.11.3	JSP 语法提要 .....	193
3.11.4	JSP 的内置对象 .....	195
3.11.5	JSP 的其他功能特性 .....	196
3.12	Telehand 系统的网络实现 .....	196
3.12.1	网络服务功能的实现 .....	196
3.12.2	操作场景的图像反馈 .....	200
<b>第 4 章</b>	<b>Java 与数据库的连接 .....</b>	<b>202</b>
4.1	JDBC 的结构和应用模式 .....	202
4.2	关系数据库和 SQL 简介 .....	204
4.3	数据库的安装和测试 .....	207
4.4	JDBC 编程基本概念 .....	210
4.4.1	装载 JDBC 驱动器 .....	211
4.4.2	与数据库的连接 .....	211
4.4.3	SQL 语句的执行 .....	213
4.4.4	数据库中返回结果的处理 .....	214
4.4.5	关闭所创建的各个对象 .....	214
4.5	JDBC API .....	214
4.5.1	DriverManager 类 .....	215
4.5.2	Connection 接口 .....	216
4.5.3	Statement 接口 .....	217
4.5.4	ResultSet 接口 .....	218
4.5.5	PreparedStatement 和 CallableStatement 接口 .....	221
4.5.6	DatabaseMetaData 和 ResultSetMetaData 接口 .....	224
4.5.7	SQLException 和 SQLWarning 类 .....	227
4.5.8	其他类和接口 .....	228
4.6	可滚动的和可修改的结果集 .....	229
4.7	高级数据类型 Blob 和 Clob .....	231
4.8	数据库的事务(Transactions) .....	233
4.9	Servlet 对数据库的操作 .....	235
4.9.1	对数据库的直接存取 .....	235
4.9.2	数据库对象的重用 .....	238
4.9.3	连接池 .....	239
4.10	一个实例: 对 Telehand 系统的网上评估 .....	242



---

第 5 章 JNI 技术 .....	250
5.1 什么是 JNI .....	250
5.2 使用 JNI 时 Java 程序的编写过程 .....	251
5.2.1 Java 程序的编写 .....	251
5.2.2 Java 程序的编译 .....	252
5.2.3 native 方法头文件的产生 .....	253
5.2.4 native 方法的实现 .....	254
5.2.5 native 共享库的生成 .....	254
5.2.6 程序的运行 .....	255
5.3 Java 与 native 之间的类型映射 .....	255
5.3.1 基本数据类型的映射 .....	256
5.3.2 引用类型的映射 .....	256
5.3.3 native 方法与实现函数之间的对应关系 .....	258
5.4 native 方法对基本数据类型的存取 .....	259
5.5 native 方法对 Java 字符串的存取 .....	259
5.6 native 方法对 Java 数组的存取 .....	261
5.6.1 基本数据数组的存取 .....	262
5.6.2 对象数组的存取 .....	264
5.7 native 方法对 Java 成员变量的存取 .....	265
5.7.1 对象成员的存取 .....	265
5.7.2 静态成员的存取 .....	267
5.7.3 类型署名和相关的 JNI 函数 .....	267
5.8 native 方法对 Java 方法的调用 .....	269
5.8.1 对象方法的调用 .....	270
5.8.2 静态方法的调用 .....	271
5.8.3 构造方法的调用 .....	272
5.8.4 JNI API: 调用 Java 方法的函数 .....	273
5.9 局部的和全局的引用 .....	274
5.9.1 局部引用、全局引用和弱全局引用 .....	275
5.9.2 与引用有关的函数 .....	277
5.10 异常的产生和捕获 .....	278
5.11 Invocation 接口 .....	281
5.12 JNI 在 Telehand 系统中的应用 .....	284
第 6 章 Java 与 VRML .....	289
6.1 什么是 VRML 和 EAI .....	289
6.2 三维图形学和 VRML 基础 .....	290
6.2.1 三维图形学的基本概念 .....	290
6.2.2 VRML 基础 .....	291

---

6.3 VRML 脚本简介 .....	294
6.3.1 几何节点 (Geometry).....	294
6.3.2 外观 (Appearance) .....	301
6.3.3 成组节点 (Grouping Node) .....	303
6.3.4 可绑定节点 (Binding Node) .....	306
6.3.5 声光节点 .....	309
6.3.6 块化和抽象化 .....	312
6.3.7 事件 (Event) .....	316
6.3.8 插补器 (Interpolator) .....	321
6.4 EAI: Java 与 VRML 的连接.....	324
6.4.1 必要的 HTML 文件.....	324
6.4.2 EAI 的工作机制.....	325
6.4.3 VRML 节点的存取.....	327
6.4.4 感知 VRML 事件的发生.....	328
6.4.5 VRML 场景的创建.....	330
6.5 EAI 在 Telehand 系统中的应用.....	331

# 第1章 绪 论

## 1.1 Java 简介

具备 Java 初步知识和 Java 基本编程经验的读者，必定对 Java 的历史、特征、优点和缺点有一定的认识。尽管如此，这里不妨再来简略回顾一下。

作为一种高级编程语言，Java 虽然诞生很晚，但它的发展、普及和应用却非常迅猛。Java 是由 Sun 公司开发和创建的，正式推出 Java 技术和 HotJava（一种支持 Java 虚拟机的 WWW 浏览器）是在 1995 年 5 月，其最早的起源和雏形则可追溯到 1991 年初。Sun 公司开发 Java 的初衷是想为消费者常用的电子设备如电视机、录放机和电话机等设计一种小型的、便于移植的软件，并为此成立了名为 Green 的项目组。当时的软件产品名称叫做 Oak，后来发现 Oak 已为别的语言所用，于是改为如今的 Java。Java 的第一个版本是在 1996 年初发布的，1.02 版在几个月之后发布。Java 发展的重大突破应归功于 1996 年初 Netscape 2.0 对它的支持。Java 的普及得益于万维网（WWW）的兴起，它的命运是与后者联系在一起的，不过，现在它的功能大大扩展了。1997 发布的 Java 1.1 具有了很多标准库，而 1998 年底发布的 Java 1.2（后改名为 Java 2）进一步扩充了标准库的内容和基本类，使其真正成为一种有广泛用途的编程语言。现在，在 Java 2 的平台基础上已发展成几个系列，包括标准版 J2SE（最新版本为 1.4.2 Beta）和企业版 J2EE（最新版本为 1.4 Beta）。

如今 Java 已广泛应用于教育、商业、通信和工程等诸多领域。Java 之所以能如此迅速传播和得到广泛应用除了因为它赶上了 IT（信息技术）时代的最好时机外，更由于它自身固有的一系列特征。这些特征包括（但不限于）下列几点。

### □ 简单性

Java 的语法比 C++ 更加简洁。它不需要输入头文件（但作为替换，要导入软件包），也没有指针、结构、联合、操作重载和虚基类等等，比 C++ 简便多了，不过它还是没有 Visual Basic 那么简便。VB 不仅语法简单，其可视化开发环境更是使应用开发几乎自动进行，而要用 Java 实现同等的功能，则需编写很多代码。不过作为一种面向对象的编程语言，Java 还是很简单的，而且现在也有不少 Java 的集成开发环境（如 Visual Café 和 Inprise JBuilder 等）可用（不过它们似乎跟不上 Java 本身的发展）。

Java 的简单性还在于它很小巧，因为 Java 最初的设计目的是用于嵌入式小电器。Java 的基本解释器和类支持只占 40KB，加上基本准库和线程支持也才 175KB（当然，如果加上 GUI 库则大得多）。

### □ 面向对象

面向对象已是现代程序设计的基本方法了。Java 的面向对象特性可与 C++ 相媲美，而最大的不同在于多重继承方面，Java 有自己更好的解决办法。

### □ 网络编程和分布式处理功能

Java 的网络编程功能很强，使用却简单而容易。它提供内涵丰富的库以支持 HTTP 和 FTP 等 TCP/IP 网络协议。通过 URL，对网络上远程对象的存取就像对本地机上的文件系统一样容易。Java 甚至使 CGI (Common Gateway Interface, 通用网关接口) 脚本编程更为容易，而且提供 Servlet 机制，使服务器端网络编程极为有效。

Java 通过客户机/服务器模式的远程方法调用，使分布对象之间的通信成为现实。

### □ 可移植性

具有良好的可移植性 (或者说与平台的无关性) 也是当初设计 Java 的主要目标之一。在这方面，Java 非常成功，最为人们所乐道。一个 Java 程序不作任何改动和不用重新编译就可运行于任何有 Java 虚拟机 (JVM, Java Virtual Machine) 的机器和操作系统上，从而做到了“编写一次，随地运行”。Java 编译器生成体系中立化的目标文件格式，即产生与具体的计算机体系和操作系统无关的字节码指令。只要有 Java (虚拟机)，编译后的代码就可以运行在任何处理器中。字节码设计的目的是为了便于在任何机器上解释和翻译为当地机器码。

### □ 解释执行

Java 程序运行前需要编译字节码形式的目标文件。字节码通过驻留在机器上的 JVM 解释来执行。JVM 实际上是一种支持 Java 程序运行的软件，它直接与操作系统和硬件打交道。正是这种机制实现了 Java 良好的跨平台性。

### □ 鲁棒性和安全性

Java 以多种方式保证程序的安全和可靠。Java 编译器能检测出很多对其他语言而言只能在运行时才能显现的问题和错误。Java 不像 C/C++ 那样使用指针对内存进行操作，因而消除了破坏内存和使数据崩溃的可能。即使 Java 程序出错，也不会对系统造成破坏和危险，任何错误的影响都在限定范围和控制中。Java 有很强的异常处理机制，Java 程序甚至能抵抗病毒的侵袭。

由于人们还期望 Java 在网络和分布式环境中使用，因此 Java 的安全性设计非常严格和可靠，其安全漏洞很少甚至没有。例如，Java 的安全特性使 Java 程序避免了自身程序空间之外的内存崩溃，避免客户通过网络浏览器对主机文件系统进行读写。可以说，Java 是当前最安全的一种编程语言。

### □ 多线程

Java 内置对多线程的支持，用 Java 作多线程编程比用别的语言容易得多。Java 的线程还可以充分利用多处理器系统的优点。Java 实现多线程的简便性是它常被用于服务器端开发的一个主要原因。

### □ 动态性

Java 程序中的类可根据需要动态地装载，甚至可以通过网络实现。另外，在 Java 中还可得到运行时的类型信息。

### □ 垃圾自动收集

Java 的自动收集垃圾机制可以检测程序中的对象是否还有用，当发现它们不再需要时，系统会自动将它们所占用的内存释放，而不需要程序对它们强制释放，因此不会有内存泄漏的问题。

Java 的上述特征也是它的优点。当然，优点这么多并不意味着 Java 是完美无缺的。Java 的运行时间影响其性能。Java 程序最终是由 Java 虚拟机解释执行的，因此其运行速度比直接使用机器指令的类似程序慢。在很多情况下，这也许不是大问题，然而在有的应用中则是一个非常严重的问题。例如，这不能满足嵌入式系统或实时系统的要求。不过，直接支持 JVM 的 Java 处理器正在研制和生产中，它们可使 Java 程序的运行速度与其他语言程序的一样快。

Java 已发展成为一种博大精深的技术了，其内容非常丰富。要掌握 Java 不是一件容易的事，而要在实际应用中综合运用 Java 的多项技术更是一项挑战。Java 除了已广泛深入到教育、商业、通信等领域之外，在工程技术和企业中的应用也越来越引人注目。

## 1.2 Java 高级应用举例：UA Telehand 系统

为了对 Java 在工程中的应用有感性认识，下面来看一个具体实例：UA Telehand 系统。该系统是作者在加拿大阿尔伯达大学（University of Alberta）计算机系开发的基于 Internet 的机器人多指手远程操作系统。利用该系统，一个远程用户（例如在中国北京的用户）可以通过 Internet 操作和控制加拿大爱德蒙顿的机器人多指手，对被抓的物体进行小范围的操作（例如移动、转动或其他任意运动）。为了便于理解后面章节的内容，下面具体介绍一下该系统的体系结构、功能和操作、实现和集成。

### 1.2.1 Telehand 系统的体系结构

Telehand 系统在体系上可以归结为三层结构，它由执行系统、服务器和客户机组成，如图 1.1 所示。

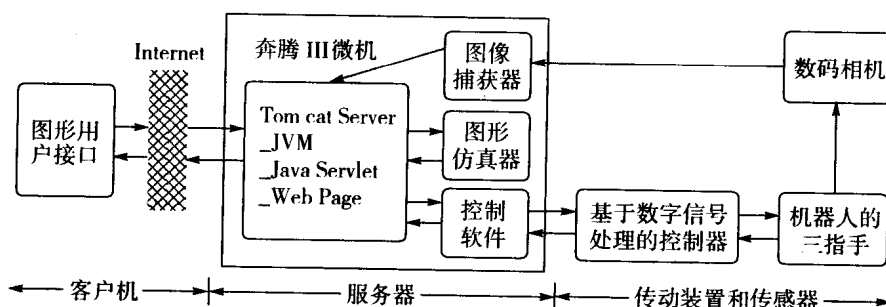


图 1.1 Telehand 系统的体系结构

#### 1. 执行系统

执行系统包括执行机构、驱动放大器和传感器等部分。执行机构是一个机器人多指手，其大小和形状类似于真人的手。它有三个手指，每个手指有三个指段、三个关节，因此整个手共有九个关节（九个自由度），各关节由直流伺服马达通过钢丝绳驱动。驱动放大器将

控制信号放大，形成伺服马达的驱动电流，控制各个关节运动。传感器包括各关节的角度传感器和一个数码相机。相机型号为 QuickCam VC，可通过并行口或 USB 口与服务器主机相连。数码相机用于对系统的操作场景进行拍照，将图像输入服务器，最终传输给远程用户，以便对多指手的操作状态进行监视。

## 2. 服务器

服务器主机用于提供网络 (Web) 服务和控制执行系统。主机是 CPU 为奔腾-III 型的微机。Tomcat 等网络服务器和 JVM 安装并运行于其上，自己开发的网络服务程序和三指手控制程序 (包括操作算法) 也驻留和运行于主机中。主机还提供有关 Telehand 系统的一些网页，包括一个供远程客户下载、用于对该系统进行远程操作的图形用户界面 (GUI, Graphic User Interface)。数码相机的驱动程序也驻留和运行在该主机中，用于抓取机器人手的图像并将图像，暂存于主机中。

另外，主机还运行一个三维图形仿真器。该仿真器根据真实的多指手建模，其大小、结构和形状与真实的手一样，主要用于演示和模仿操作任务的执行。由于多指手对物体进行灵巧操作的复杂性，有时很难想象一个操作在执行过程中和完成之后手指/物体位形的变化，很难预知物体是否会滑落、手指之间会否互相干涉和碰撞、手指运动空间相对于操作任务而言是否足够等等。因此在真实的多指手执行操作之前有必要先动态仿真一番，以便预览期望的操作能否完成和是否安全。仿真器的功能还可以扩展为操作任务的图形输入。

在主机上还安插了基于 DSP (数字信号处理)、执行 PID 算法的运动控制板，其控制信号输出到多指手的驱动放大器中。

为了能让远程客户使用 Telehand 系统，服务器主机必须连接在 Internet 上，并将网络服务器启动。而要实际操作多指手时，还必须将放大器的电源打开。一般情况下，它们都是关闭的，因此用户不能随便试用。

## 3. 客户机

客户机是为远程用户使用的。任何一个与 Internet 相连并安装支持 Java 的浏览器 (例如 IE-Internet Explorer 或 Netscape) 的微机或工作站都可以充当客户机。在客户机上下载 Telehand 的 GUI 页面后即可对系统进行操作了。

Telehand 系统的图形用户界面如图 1.2 所示。它由三部分组成：左上部分显示传过来的数码相机捕获的图像，刷新率为每秒一帧；右上部分则是图形仿真器；下面区域用作操作时的输入输出。下面的区域又可划分为三个区：右边用于设定系统状态和发布操作命令，中间为物体选择和操作任务输入区，左边则为系统状态显示区。应该指出，为了显示图形仿真器，客户机上的浏览器要有 VRML (Virtual Reality Modeling Language, 虚拟现实造型语言) 插件，如 Cosmo Player 或 Cortona 等。当鼠标指针放置在仿真器区域时，单击鼠标的左键或右击便可以改变仿真器的视点和视角。

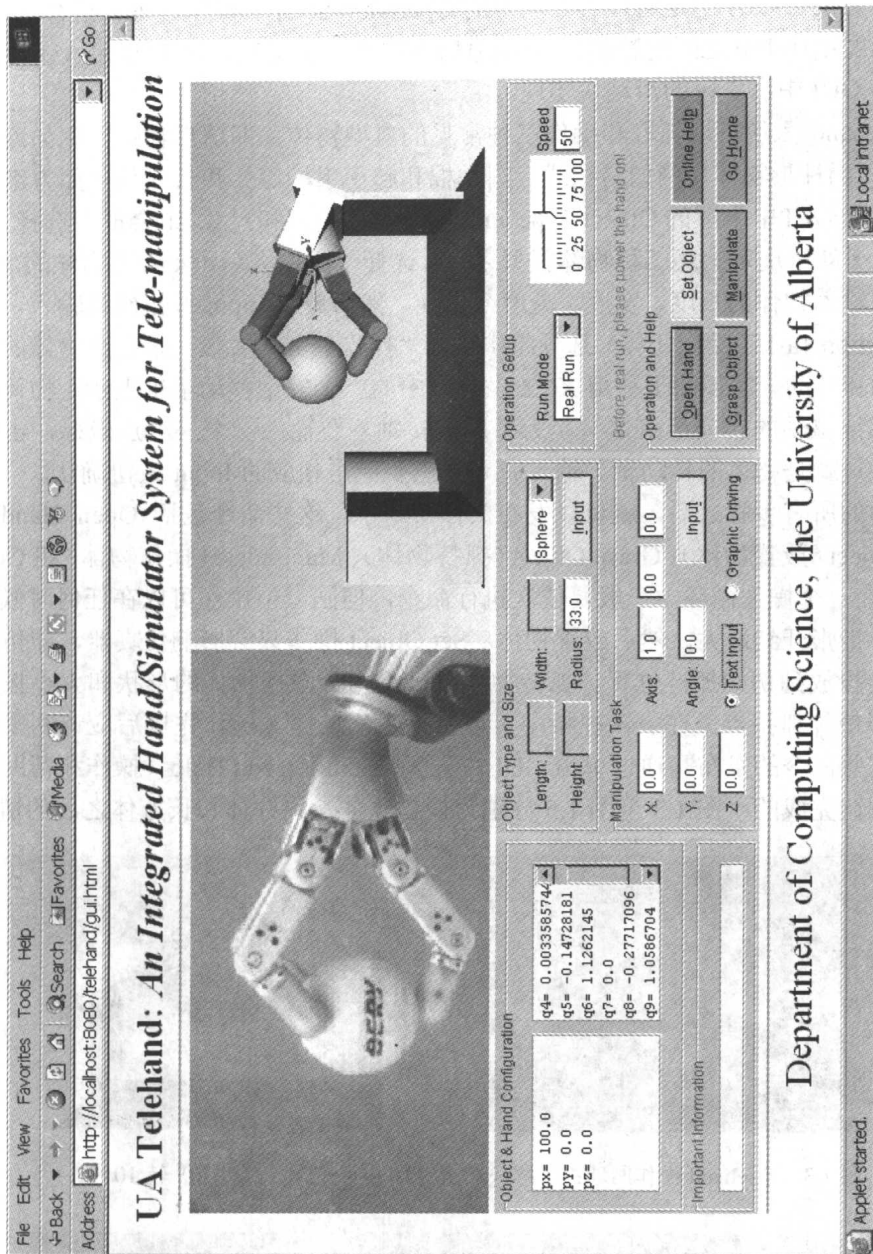


图 1.2 UA Telehand 系统的图形用户界面 (GUI)

### 1.2.2 Telehand 系统的功能和操作

如上所述, Telehand 系统用于通过 Internet 利用一个机器人多指手对被抓物体进行远程操作。系统的使用模式有两种, 即仿真 (Simulation) 和实际操作 (Real Run)。在仿真模式中, 真实的多指手不会运动 (无论其电源是否打开), 只有仿真器在受控制和运动; 而在实际操作模式中, 当电源开启后, 真实的多指手会被驱动和控制, 同时仿真器也会有动作。

模式的选择和切换在 GUI 中是通过右下区中的 Operation Setup (操作设置) 组合框实现的。物体的操作速度用 0~100 之间的整数表示(为 0 时多指手不会运动, 100 时以最大速度运动), 速度的调节用 GUI 中右下区的滑动块实现。

目前 Telehand 系统可以抓取和操作三种常见的典型物体, 即球形物体 (例如网球、乒乓球和鸡蛋)、圆柱形物体 (例如易拉罐、圆木棍和起子手柄等) 和长方体 (如方纸盒和方木块等)。从位于 GUI 中下区的 Object Type and Size (物体类型和尺寸) 组合框中选择 Sphere、Cylinder 或 Box 即可选定上述三种物体类型之一。选择了类型之后, 要给出物体的尺寸, 根据不同的类型输入物体的长度、宽度、高度或半径, 然后单击 Input 按钮确认输入。下一步是在 Manipulation Task (操作任务) 选项组中输入“操作任务”参数, 来定义物体的期望运动, 包括移动和转动。物体的平移用 XYZ 坐标系中的一个三维移动量来表示; 物体的转动由一个转轴和绕该轴的转动量来定义。为了确定转轴, 要输入一个三维点坐标, 由坐标原点到该点的矢量就是物体的转动轴。输入完毕后也要单击相应的 Input 按钮确认。

完成上面的所有步骤之后, 就可以发布操作命令了。这些命令包括 Open Hand (张开手指)、Set Object (放置物体)、Grasp Object (抓持物体)、Manipulate (操作物体) 和 Go Home (回零位)。在一个操作循环中一般要依次执行命令, 但回零位操作可以在任何时候执行。首先要将各手指张开到最大, 然后放置物体。Set Object 操作只会影响仿真器, 即仿真器中的相应物体会投放到初始抓持位置。抓持物体时, 手指会根据物体的形状和大小按内置的算法自动选择最优抓持点。在一次抓持中可以执行多次操作。操作完毕后应该回零位, 使手指伸直。这些命令的发布是通过单击 GUI 右下区 Operation and Help (操作和帮助) 选项组中相应的按钮完成的。图 1.3 至图 1.5 分别为操作球体、圆柱体和长方体之后的情形。

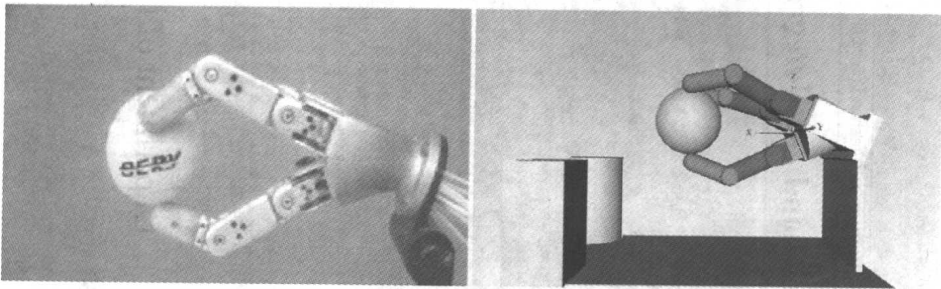


图 1.3 Telehand 操作网球: 平移 (10, 10; 10), 并绕轴 (1, 1, 1) 转 10 度

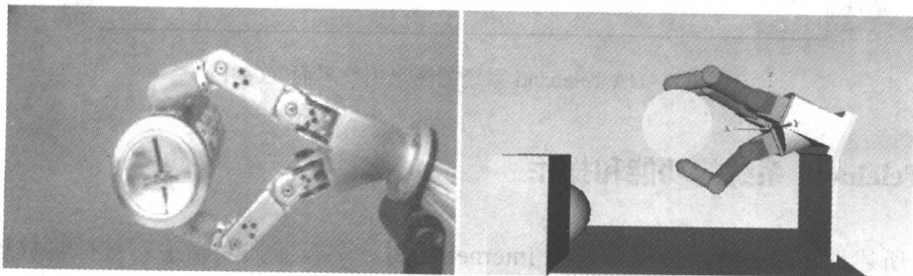


图 1.4 Telehand 操作易拉罐: 绕 Y 轴 (0, 1, 0) 转 15 度



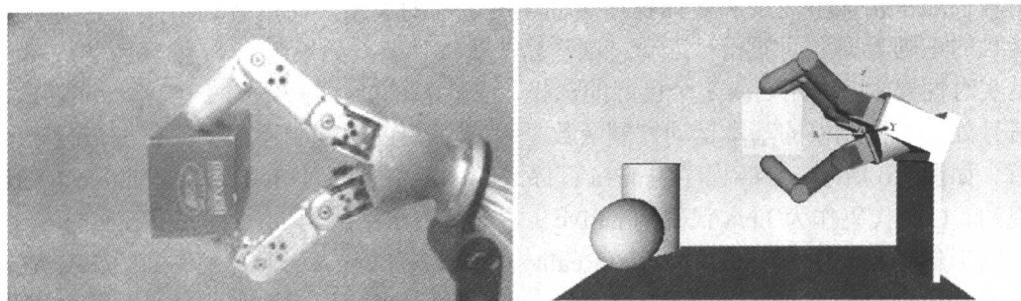


图 1.5 Telehand 操作方纸盒：平移  $(-10, 10, 0)$  mm

每次操作之后都会在 Object & Hand Comfiguration (物体和手的位形) 组合框中显示系统的位形信息, 包括物体的位置 (三个坐标值) 和姿态 (三个角度值), 以及各手指关节的角度值。另外, 系统还可能在 Important Infomation (重要信息) 框中显示警告或提示信息。例如, 在操作顺序不对 (如没有张开手指就放置物体、还没有放置物体就进行抓持、没有抓持物体就想执行操作任务等) 的情况下, 系统不会动作, 只显示必要的提示。

此外, GUI 中还有个 Online Help 按钮。单击之后会为用户弹出有关 Telehand 系统的描述和使用指南。

### 1.2.3 Telehand 系统的实现和集成

Telehand 系统的实现涉及多种编程语言和技术, 包括 Java、C++和 VRML, 它们之间取长补短, 分工协作。其软件结构包括三大部分: 网络服务和 GUI、操作和控制, 以及图形仿真, 如图 1.6 所示。

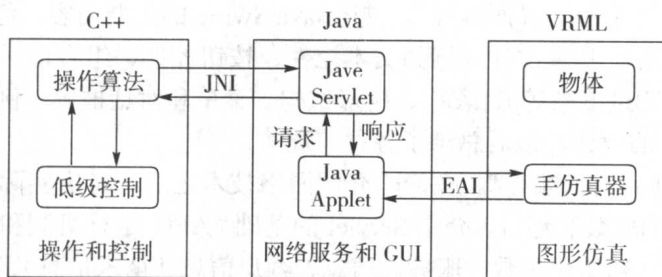


图 1.6 Telehand 系统的软件结构

其中, 核心部分网络服务和 GUI 用 Java 编写, 并由 Java 集成和管理其他部分。网络服务用 Java Servlet 实现, 而 GUI 则设计成 Applet, 用 Java Swing 实现。它们之间的通信为 HTTP 请求-响应模式。

多指手的操作和控制需要有很强的实时性, 要求算法快、程序运行效率高, 因此操作算法和底层控制用 C++实现。多指手操作的目的是要将被抓物体从一个初始位形驱动到期望的位形, 这就要求操作算法根据给定的前后位形控制各手指、各关节的运动。Telehand 系统的操作算法同时考虑了多指手运动学和接触运动学。前者通过多指手雅可比矩阵和抓