第三章 构建WEB应用技术

- 章节目标
  - 1. 了解WEB应用模型，掌握四层架构模型。
  - 2. 掌握HTML和XML的特点，能够使用HTML和XML做网页。了解脚本技术。
  - 3. 了解CGI的工作原理和API的特点。掌握Servlet的生命周期和处理事件的四种方法。
  - 4.能够写简单的JSP页面，了解其特点。

华中科技大学 第三章 构建WEB应用技术

第三章 构建WEB应用技术

- 章节目标
  - 5.掌握Web Service的体系架构和SOAP，UDDI，WSDL的概念和基本功能。
  - 6. 掌握J2EE中各容器的功能，了解其构件技术、服务技术和通信技术。
  - 7.了解分布式对象系统。

本章目录

3.1 WEB应用模型

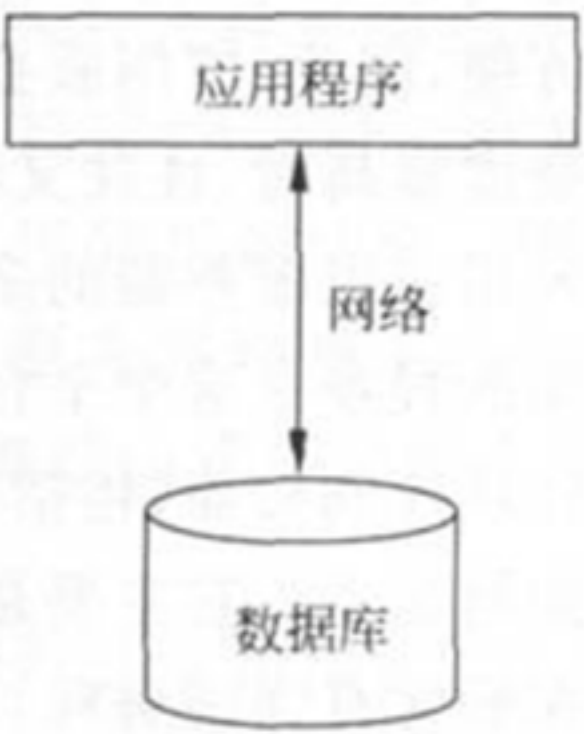
3.1.1 WEB应用模型概述

- WEB应用的发展：

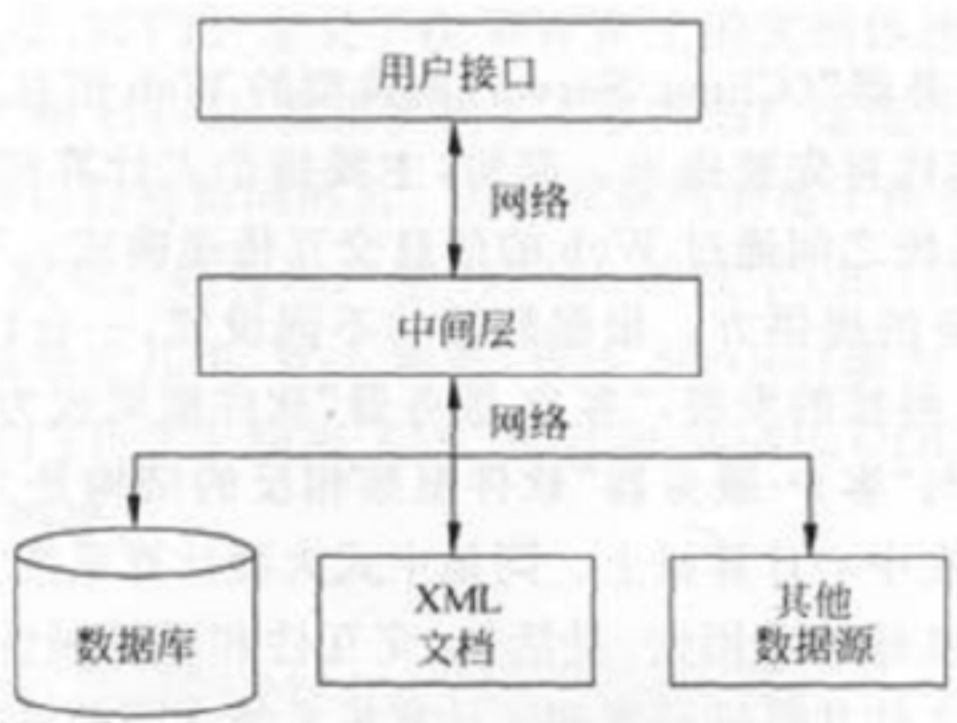
  静态HTML页面--〉动态WEB页面

- 典型WEB架构：客户/服务器 架构

3.1.2 两层架构WEB应用模型



3.1.3 三层架构WEB应用模型



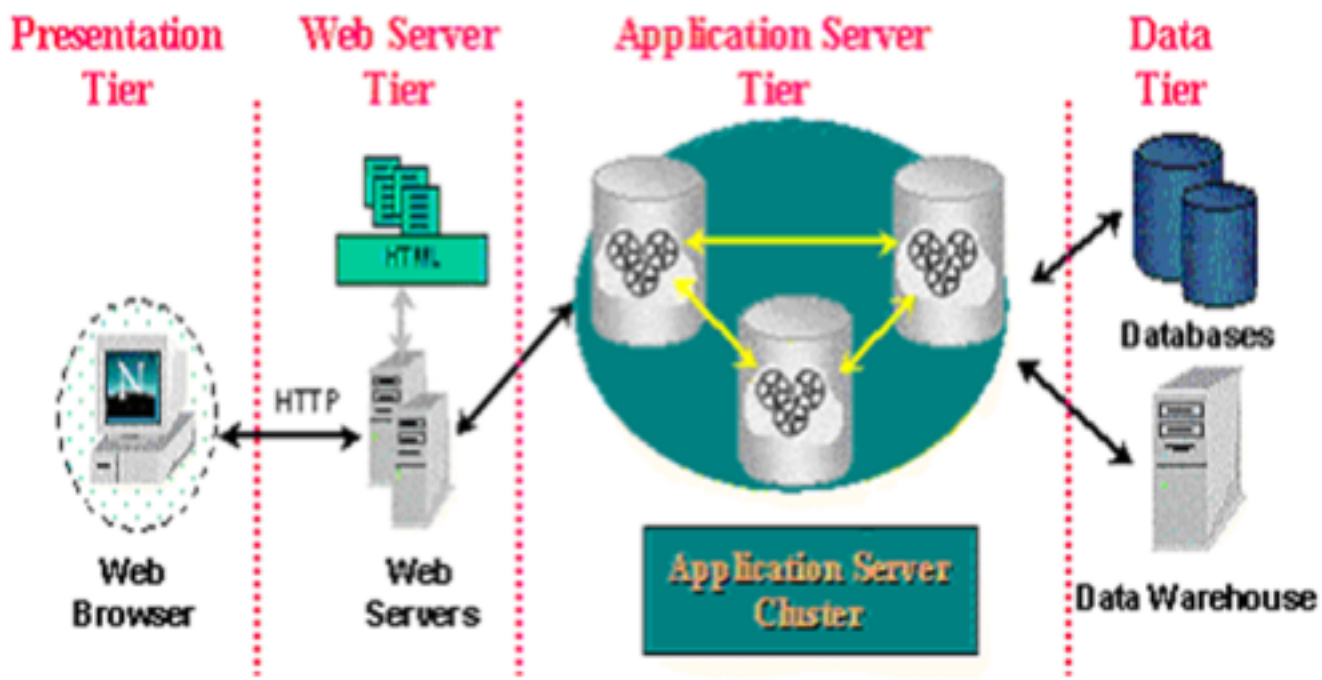
3.1.4 N层架构WEB应用模型

- 概述
- 四层架构模型
- N层架构系统的可规模化
- N层架构应用系统的优势

3.1.4 N层架构WEB应用模型 –概述

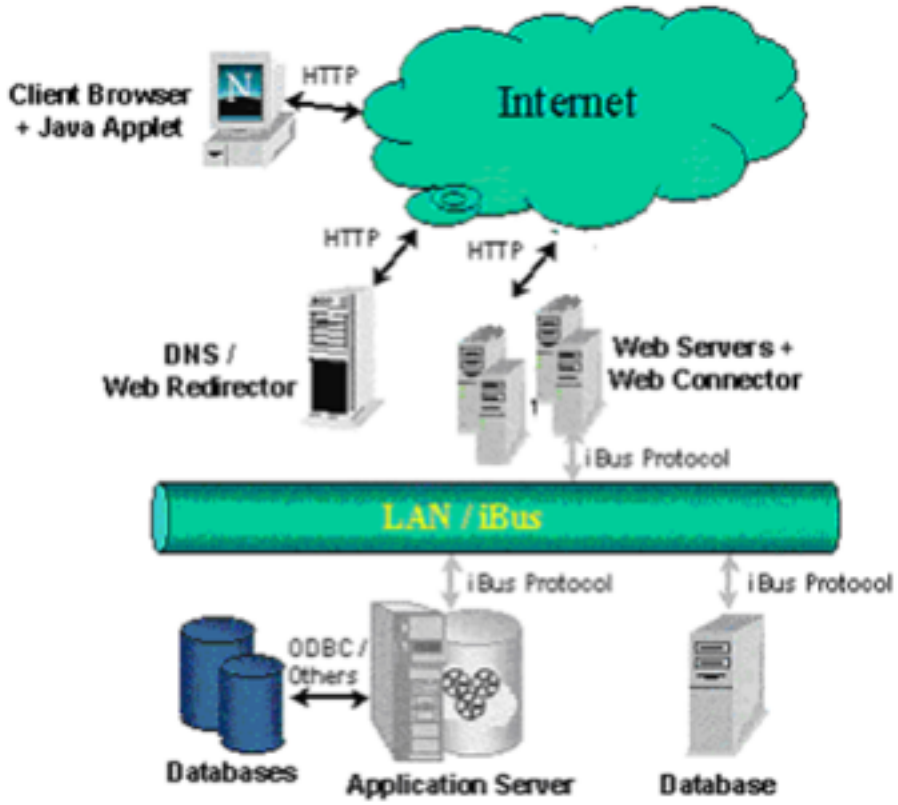
- N层架构的目的是要在HTTP协议上解决连结保持、状态转化和事务完整性等问题。
- N层架构的核心是要提供软件系统的可规模化（scalability）。

3.1.4 N层架构WEB应用模型 --四层架构模型



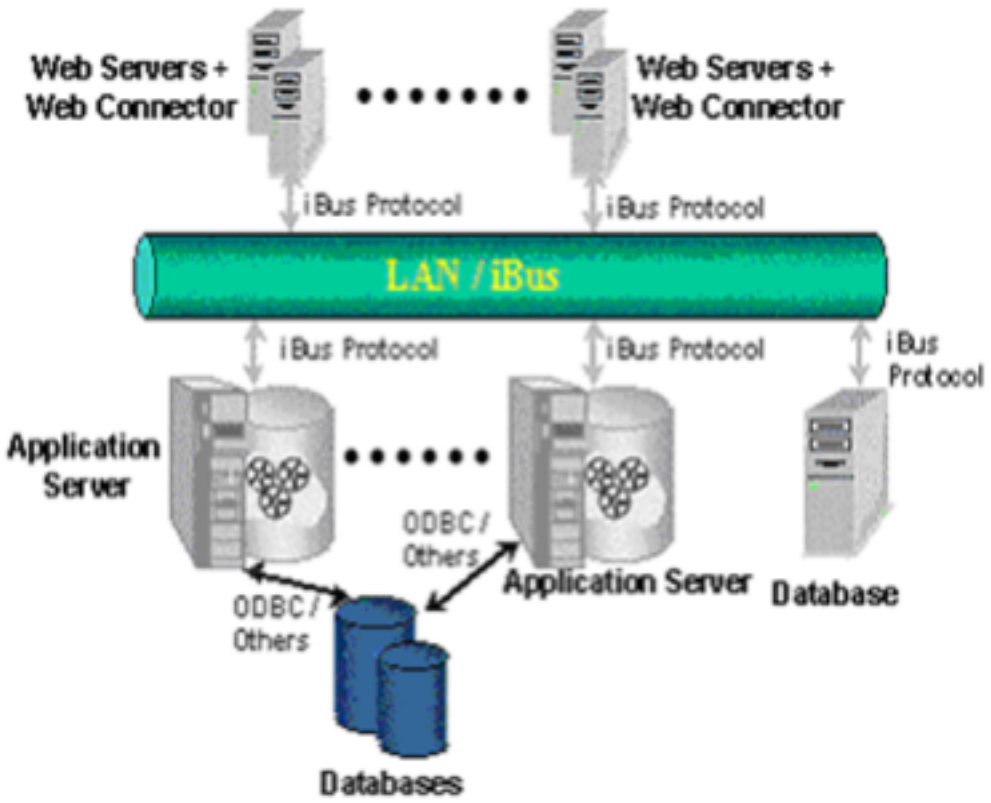
3.1.4 N层架构WEB应用模型 --可规模化

- 因特网上的可规模化



3.1.4 N层架构WEB应用模型 --可规模化

- 企业内部网上的可规模化



3.1.4 N层架构WEB应用模型 --优势

- 信息引擎架构
- 全面面向Internet
- 系统性能可规模化
- 系统功能可规模化
- 系统维护方便
- 系统具有企业应用强度

本章目录

3.2 客户端技术

- 3.2.1 HTML
- 3.2.2 脚本技术
- 3.2.3 XML

3.2.1 HTML

- HTML是什么
- HTML文件大致结构
- HTML标记
- HTML的特点

3.2.1 HTML -- HTML是什么

- HTML是Hypertext Markup Language（超文本标记语言）的缩写
- HTML是构成WEB页面的主要工具，是用来表示网上信息的符号标记语言。
- SGML： Standard Generalized Markup Language
- HTML是SGML的一个简化的实现

3.2.1 HTML -- HTML文件大致结构

```
<HTML>
        <HEAD>
          <TITLE>网页标题</TITLE>
        </HEAD>
        <BODY>
            网页的内容
        </BODY>
</HTML>
```

3.2.1 HTML -- HTML标记

标记写法需要遵循以下规则：

- 任何标记皆由"<"及">"所围住，如 <P>。
- 标记名与小于号之间不能留有空白字符。
- 某些标记要加上参数，某些则不必。如 Hello
- 参数只可加于起始标记中。
- 在起始标记之标记名前加上符号"/"便是其终结标记，如 。
- 标记字母大小写皆可。

华中科技大学 第三章 构建WEB应用技术

3.2.1 HTML -- HTML标记

- 文件标记：<HTML>，<HEAD>，<TITLE>，<BODY>
- 排版标记：<P>，
，<CENTER> 等
- 字体标记：<H1>，<H2>，<H3>，<H4>等
- 表格标记：<TABLE>，<TR>，<TD> 等
- 表单标记：<FORM>，<INPUT> 等
- 连结标记：<A>
- 框架标记：<FRAMESET>，<FRAME> 等

3.2.1 HTML -- Html的特点

特点：

- 简单易学；
- 解释执行，无需编译；
- 独立于平台；

3.2.2 脚本技术

脚本语言的优势：

- 开发快速
- 部署简便
- 能方便的与已有技术集成
- 易于学习和使用
- 动态代码

3.2.2 脚本技术

- 3.2.2.1 Java Script
- 3.2.2.2 VB Script

3.2.2.1 Java Script

- 简介
- 例题
- 特点
- 与Java比较

3.2.2.1 Java Script --简介

- 由 Netscape公司开发
- 基于对象和事件驱动的编程语言
- 是一种解释执行的编程语言

3.2.2.1 Java Script --例题

```
<script Language="JavaScript">
var msg = "将分辨率设为800x600以上，获得最佳效果！";
var interval = 100
var spacelen = 120;
var space10="  ";
var seq=0;
function Scroll() {
len = msg.length;
window.status = msg.substring(0, seq+1);
seq++;
if ( seq >= len ) {
seq = 0;
window.status = '';
window.setTimeout("Scroll();", interval );
}
else
window.setTimeout("Scroll();", interval );
}
Scroll();
</script>
```

3.2.2.1 Java Script --特点

- 解释性编程语言
- 基于对象的语言
- 基于事件驱动的语言
- 良好的安全性
- 跨平台性

3.2.2.1 Java Script – 与Java比较

- 基于对象和面向对象
- 解释和编译
- 弱变量和强变量
- 代码格式
- 嵌入方式
- 动态链结和静态链结

3.2.2.2 VB Script

- 简介
- 事件处理程序
- 特点

3.2.2.2 VB Script --简介

- 是微软创建的一种脚本语言
- 全称是Visual Basic Scripting Edition
- 实际上是Microsoft Visual Basic的一个子集，是基于Visual Basic的脚本编写语言

3.2.2.2 VB Script -- 事件处理程序

- 事件过程
- 内联事件处理
- FOR/EVENT属性方式

3.2.2.2 VB Script -- 特点

- 简单易学
- 安全性好
- 可移植性好

3.2.3 XML

- 诞生和发展
- 基本语法和实例
- 主要相关技术
- 特点
- 使用前景
- 开发工具

3.2.3 XML --诞生和发展

- 1986年，SGML成为国际标准规范。
- SGML语言文件组成：语法定义、文件类型定义DTD（Definition Type Document）和文件实例 。
- 1991年，蒂姆·伯纳斯·李定义了HTML语言的第一个规范，之后成为符号化语言规范。
- HTML的DTD被固定。
- 1996年，提出了XML（Extensible Markup Language）语言草案。
- 1998年，W3C正式发布了XML 1.0标准。
- XML是SGML的一个简化子集。XML有DTD 。

3.2.3 XML --基本语法和实例

messager.dtd文件：

- <!ELEMENT messager (service*)>
- <!ELEMENT service (contactlist*)>
- <!ATTLIST service name CDATA #REQUIRED>
- <!ELEMENT contactlist (contact*)>
- <!ELEMENT contact (#PCDATA)>

3.2.3 XML --基本语法和实例

messager.xml文件：

```
<?xml version="1.0" ?>
  <!DOCTYPE messager SYSTEM "./messager.dtd">
    <messenger>
      <service name=".NET Messenger Service">
        <contactlist>
          <contact>Jack@sina.com</contact>
        </contactlist>
      </service>
    </messenger>
```

3.2.3 XML --主要相关技术

- 1. XML Parser
- 2. 定义XML数据结构的技术
- 3. 显示和打印XML数据的技术
- 4. XML数据结构转换技术

3.2.3 XML –特点

- 标记的可扩展性
- 数据存储和数据显示的分离
- 具有自描述性

3.2.3 XML --使用前景

- 商务的自动化处理
- 信息发布
- 智能化的Web应用程序和数据集成

3.2.3 XML --XML的开发工具

- Notepad
- Microsoft XML Notepad
- Visual InterDev
- Microsoft XML Tree Viewer
- Microsoft XML Validator
- Microsoft XSL Debugger

本章目录

3.3 服务器端技术

- 静态 --〉动态
- SSI（Server Side Includes）
- 1993年，CGI 1.0 发布。
- 1994年，发明PHP
- 1996年，ASP技术引入
- 1997年，Servlet技术问世
- 1998年，JSP技术诞生

3.3 服务器端技术

- 3.3.1 CGI
- 3.3.2 API
- 3.3.3 Servlet
- 3.3.4 JSP

3.3.1 CGI

- 概述
- 工作原理
- CGI应用的实现

3.3.1 CGI –概述

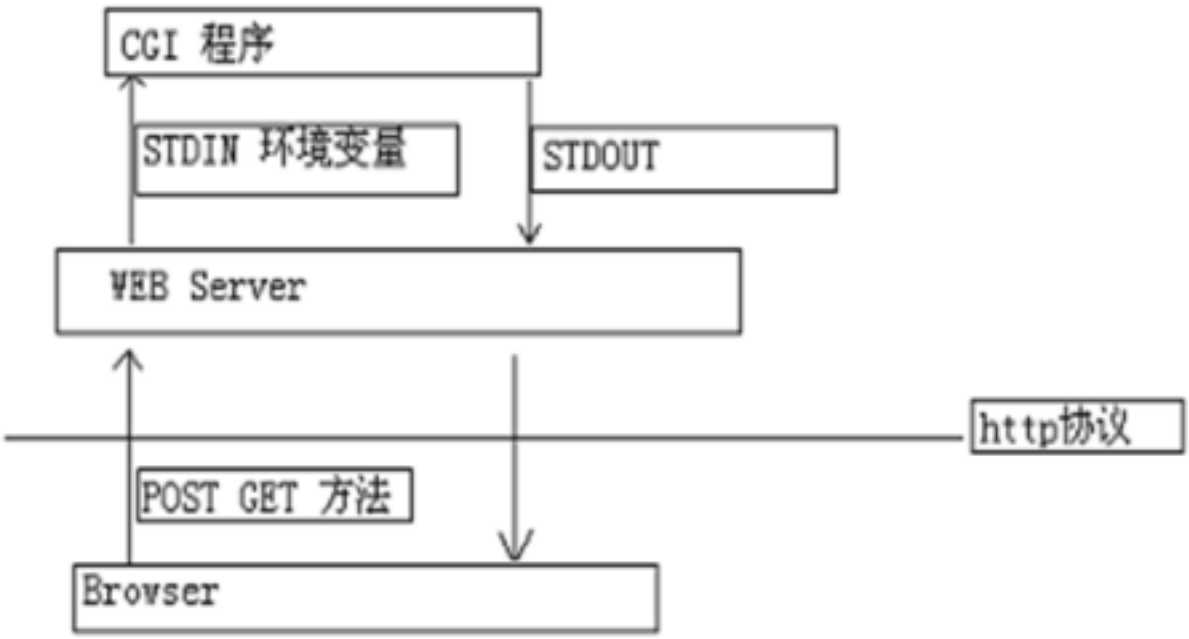
- CGI (Common Gateway Interface)： 通用网关接口
- 通过CGI，Web服务器能够执行应用程序，并将应用程序的输出，如文字、图形、声音等传递给一个Web浏览器
- CGI程序语言:C、Shell、Perl和Visual Basic 等

华中科技大学 第三章 构建WEB应用技术

3.3.1 CGI –概述

- CGI程序在UNIX系统中放在cgi-bin子目录下,在Windows系统中放在cgi-win下。
- CGI程序两种调用方式：
  （1）URL调用：
  http://gjy.sic.o.ml.org/cgi-win/cgi.exe
  （2）HTML里的Form调用

3.3.1 CGI --工作原理



3.3.1 CGI -- CGI应用的实现

- 编写交互式页面
- 编写CGI程序

3.3.2 API

- 概述
- 优缺点

3.3.2 API –概述

- API（Application Program Interface）：应用程序接口
- API解决多线程问题，而CGI无法实现多线程
- API使用动态链接库
- 只能用C语言编写
- 两种API：Netscape公司的NSAPI 和Microsoft的ISAPI

3.3.2 API --优缺点

- 优点：执行速度快
- 缺点：
  （1）NSAPI 及 ISAPI 对网站平台的依赖型太高
  （2） NSAPI 及 ISAPI 只提供 C 程序语言的界面
  （3）在变量数据的处理上要特别小心

3.3.3 Servlet

- 简介
- 生命周期
- 四种方法
- 特点

3.3.3 Servlet –简介

- Servlet就是Java Servlet
- Servlet是一个Java类
- Servlet运行于由Servlet引擎所管理的Java虚拟机中，被来自Web客户机的请求所唤醒并用来处理请求
- Servlet的开发工具可J2SDK开发工具箱
- 主要有两个软件包：Javax.servlet包和Javax.servlet.http包

3.3.3 Servlet --生命周期

- Servlet的生命周期：
  （1）初始化时期：**init()**方法
  （2）执行时期：**service()**方法
  （3）结束时期：destroy()方法

3.3.3 Servlet -- 四种方法

HttpServlet类中常用的四种方法：

- doGet( )方法
- doPost( )方法
- doPut( )方法
- doDelete( )方法

3.3.3 Servlet -- 特点

- 高效
- 方便
- 功能强大
- 可移植性好
- 节省投资

3.3.4 JSP

- 概述
- 语法
- 实例
- 技术特点
- 与其它动态网页技术比较

3.3.4 JSP --概述

- JSP： Java Server Pages
- 用于创建可支持跨平台及跨Web服务器的动态网页
- 简单的说，一个JSP网页就是在HTML网页中包含了能够生成动态内容的可执行应用程序代码。
- JSP将应用程序逻辑和页面显示分离

3.3.4 JSP --语法

- JSP的基本语法：
（1）JSP指示：JSP页面指示； 语言指示；包含指示； 标识库指示
（2）JSP标识：核心标识为 jsp:useBean ；jsp:setProperty； jsp:getProperty
（3）脚本元件：脚本在<%和%>标志中被描述，常用的脚本元件包括表达式和声明

3.3.4 JSP --实例

```
<HTML>
<%@ page language=="java" imports=="com.wombat.JSP.*" %>
<H1>Welcome</H1>
<P>Today is </P>
<jsp:useBean id=="clock" class=="calendar.jspCalendar" />

<ul>
<li>Day: <%==clock.getDayOfMonth() %>
<li>Year: <%==clock.getYear() %>
</ul>
```

3.3.4 JSP --实例（续）

- <% if (Calendar.getInstance().get(Calendar.AM_PM) ==== Calendar.AM) { %>
- Good Morning
- <% } else { %>
- Good Afternoon
- <% } %>

- <%@ include file=="copyright.html" %>

- </HTML>

3.3.4 JSP --技术特点

- 将内容的生成和显示进行分离
- 强调可重用的组件
- 采用标识简化页面开发

3.3.4 JSP --与其它动态网页技术比较

- 与传统的CGI方式相比
- JSP和ASP相比
- JSP和纯Servlet相比
- JSP和JavaScript相比

本章目录

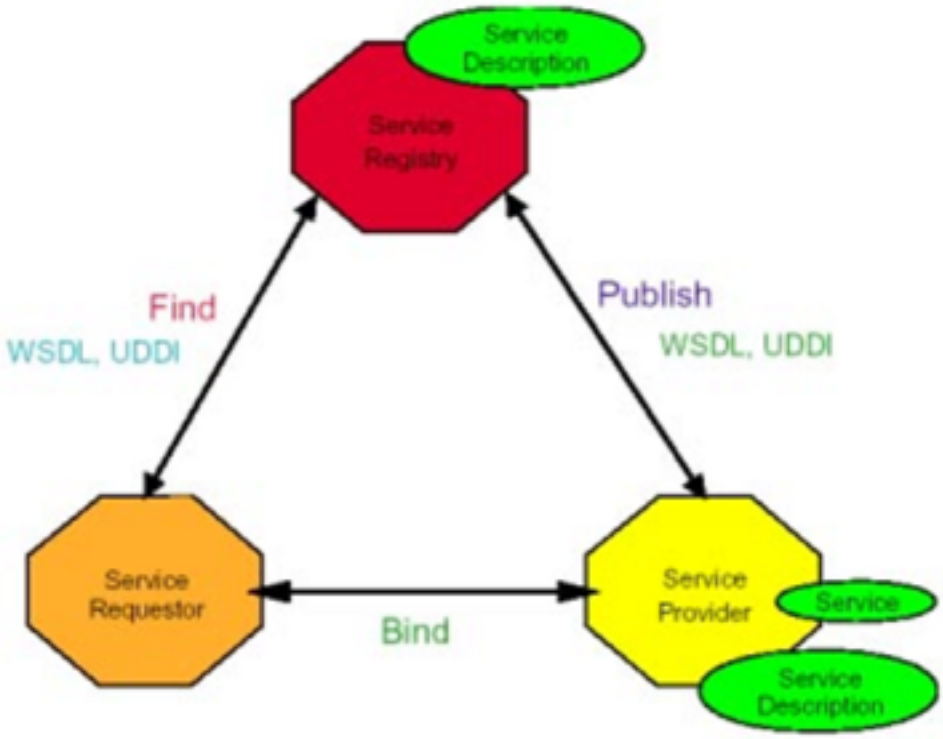
3.4 Web service

- 简介
- 体系架构
- 整体架构
- SOAP
- UDDI
- WSDL
- 特点

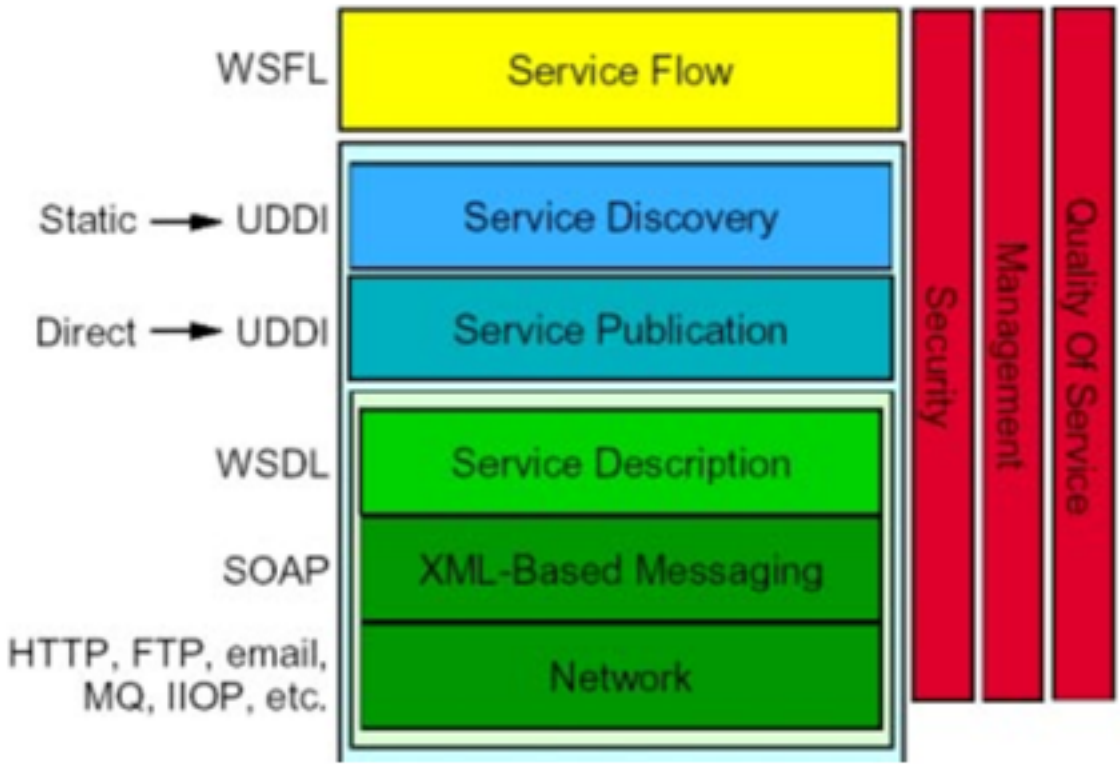
3.4 Web service --简介

- Web服务使用基于XML的消息处理作为基本的数据通讯方式
- 它消除使用不同组件模型、操作系统和编程语言的系统之间存在的差异
- 它使异类系统能够作为单个计算网络协同运行

3.4 Web service --体系架构



3.4 Web service –整体架构



3.4 Web service -- SOAP

- SOAP（Simple Object Access Protocal)：简单对象访问协议
- 是一种基于XML的不依赖传输协议的表示层协议，用来在应用程序之间方便地以对象的形式交换数据

- SOAP包括三个部分：SOAP封装结构；SOAP编码规则；SOAP RPC表示

3.4 Web service -- UDDI

- UDDI(Universal Description, Discovery and Integration)：统一描述、发现和集成协议
- 是一套基于Web的、分布式的、为Web Service提供的信息注册中心的实现标准规范。
- 核心组件是UDDI商业注册，它使用一个XML文档来描述企业及其提供的Web Service 。

3.4 Web service -- WSDL

- WSDL(Web Services Description Language)：Web服务描述语言
- 它把网络服务定义成一个能交换消息的通信端点集
- 可重用抽象定义：
  消息，是需要交换的数据的抽象描述。
  端口类型，是操作的抽象集合。
- 定义网络服务时使用的元素：类型；消息；操作；端口类型；绑定；服务。

3.4 Web service –特点

- 完好的封装性
- 松散耦合
- 使用协约的规范性
- 使用标准协议规范
- 高度可集成能力
- 开放性

本章目录

3.5 J2EE

- 简介
- 多层Web框架技术
- 构件技术
- 服务技术
- 通信技术

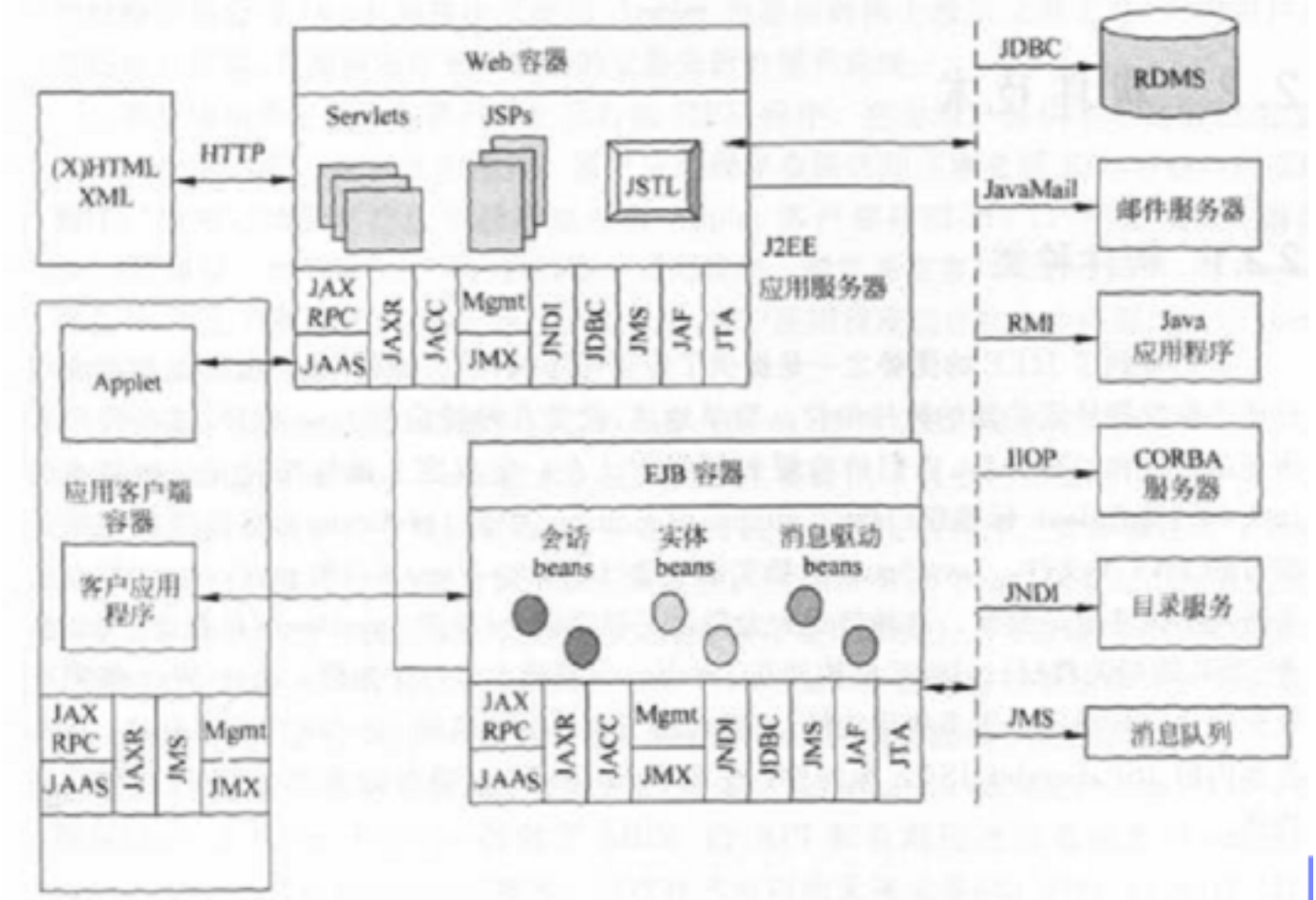
3.5 J2EE --简介

- J2EE是纯粹基于Java的解决方案
- 1999年，Sun正式发布了J2EE的第一个版本。
- IBM的WebSphere、BEA的WebLogic
- 2003年时，J2EE升级到1.4版，其中，Servlet V2.4、JSP V2.0和EJB V2.1

3.5 J2EE --多层Web框架技术

- J2EE 1.4 由以下程序容器组成：
  （1）EJB容器
  （2）Web容器
  （3）应用客户端容器
  （4） Applet容器

- J2EE 1.4多层次结构Web程序框架图如下一页图所示。



华中科技大学 第三章 构建WEB应用技术

3.5 J2EE --构件技术

- J2EE的客户
- Web构件
- 企业JavaBeans(EJB)构件

3.5 J2EE --服务技术

- 命名技术(JNDI)
- 数据连接技术(JDBC)
- 数据事务技术
- 安全技术
- 连接框架技术
- Web服务技术

3.5 J2EE --通信技术

- Web协议
- 远程方法调用
- 对象管理组协议
- Java通信服务技术
- Java消息技术和邮件技术

本章目录

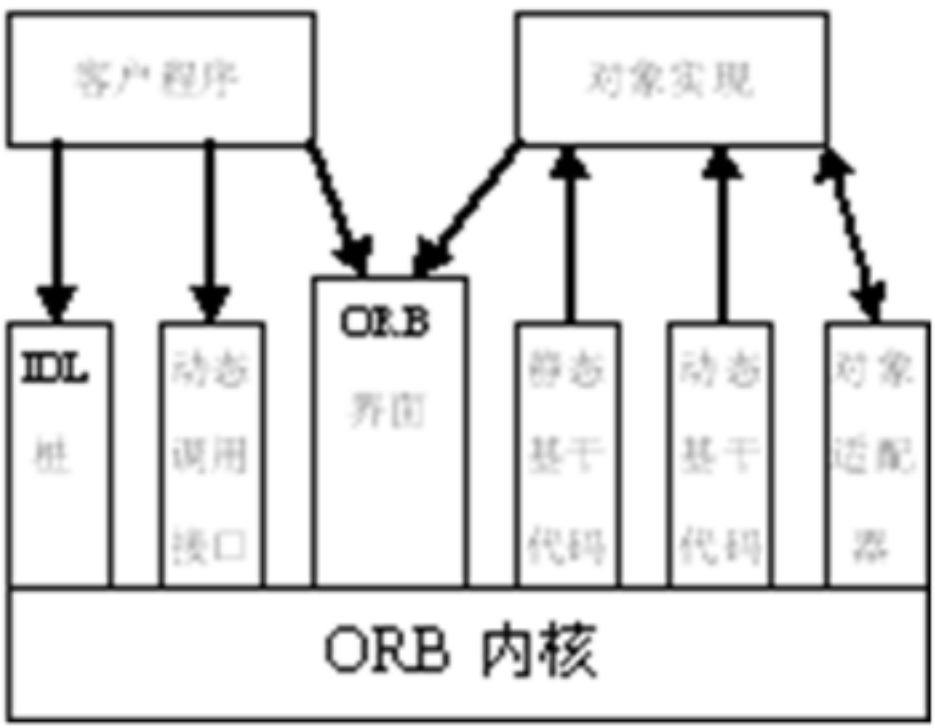
3.6 分布式对象系统

- 分布式对象技术
- CORBA体系结构
- 分布式对象技术的比较
- 分布对象技术的特点
- 与Web技术的集成

3.6 分布式对象系统-- 分布式对象技术

- 分布对象又被称为组件，组件是一些独立的代码的封装体
- 分布式组件对象标准极大地推动了以异构环境下协同工作为目标的虚拟环境研究
- 三大分布式组件对象标准：
  CORBA； DCOM； RMI

3.6 分布式对象系统-- CORBA体系结构



3.6 分布式对象系统–分布式对象技术的比较

- COM与CORBA的比较
- CORBA与RMI和DCOM的比较

3.6 分布式对象系统--特点

- 集成性
- 可用性
- 可扩展性

3.6 分布式对象系统--与Web技术的集成

- 它可以消除CGI的瓶颈，使客户端能直接调用服务器上的方法
- 可以让请求对象运行在多个服务器上
- CORBA 可以采用分布式对象结构扩充 Java

本章目录

3.7 Web开发技术的未来

- Web的未来是语义化的Web（Semantic Web）
- 2004年2月，W3C宣布RDF和OWL标准正式成为W3C的建议方案，这标志着语义化Web的大厦已经破土动工。