

CERP软件开发流程介绍

徐汉川

2004年12月30日



主要内容

□ 系统开发框架介绍

- ❖ J2EE简介
- ❖ MVC设计模式
- ❖ CERP系统框架
 - 系统环境
 - 框架运行流程
 - 框架元素构成
 - 权限控制

□ 软件开发流程

- ❖ 开发流程图
- ❖ 系统界面的设计与实现
 - 界面元素构成
 - 典型界面演示
 - 界面设计原则
 - 界面的开发
- ❖ 系统类的设计与实现
- ❖ 应用服务器的使用简介
- ❖ 实例演示

□ 课后作业



主要内容

❑ 系统开发框架介绍

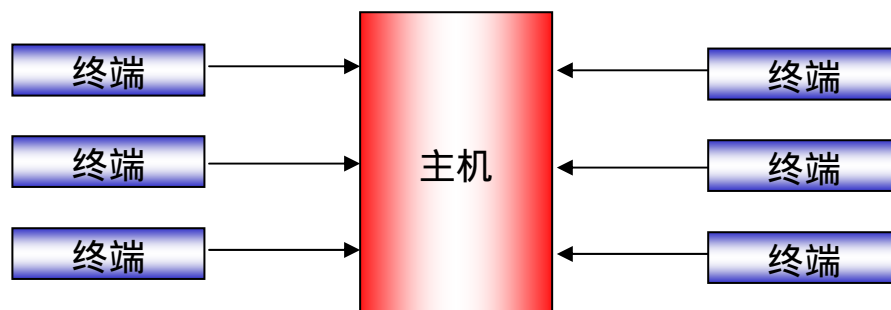
- ❖ J2EE简介
- ❖ MVC设计模式
- ❖ CERP系统框架
 - 系统环境
 - 框架运行流程
 - 框架元素构成
 - 权限控制

❑ 软件开发流程

- ❖ 开发流程图
- ❖ 系统界面的设计与实现
 - 界面元素构成
 - 典型界面演示
 - 界面设计原则
 - 界面的开发
- ❖ 系统类的设计与实现
- ❖ 应用服务器的使用简介
- ❖ 实例演示

❑ 课后作业

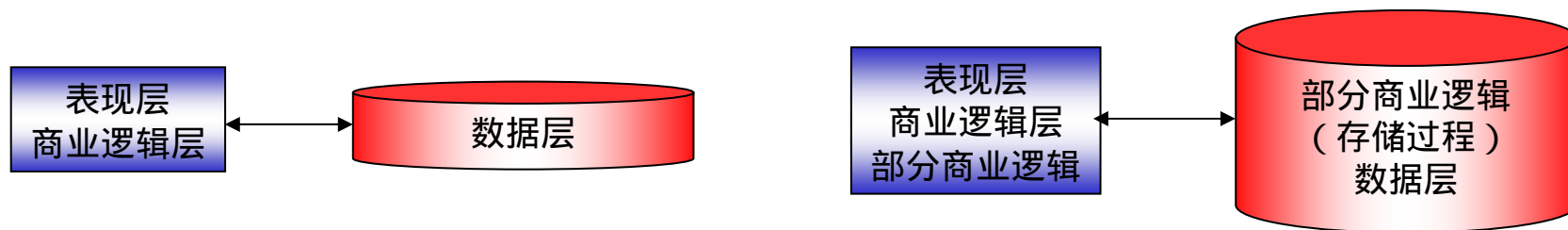
◆主机系统或单层结构：主机/终端模式



- ❑ 所有计算和操作均在主机完成；终端只能输入命令和显示结果，没有存储器
- ❑ 数据访问、表示和业务逻辑在一个应用中，功能紧耦合在一起
- ❑ 代码复用、代码可维护性、代码的修改十分困难
- ❑ 应用不是分布式的，不具有可伸缩性

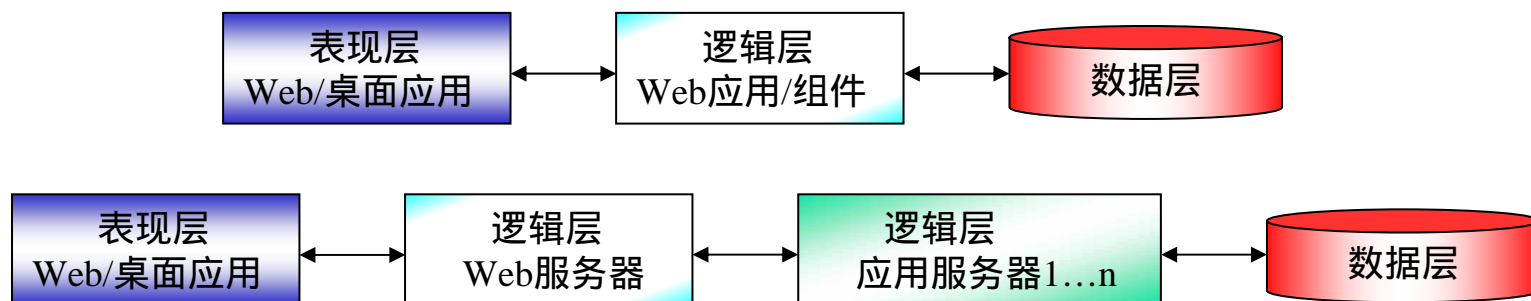
J2EE简介：传统应用程序模型2

◆ 双层客户/服务器结构：“胖客户端”



- ❑ 服务器的数据库必须同每一个活动的客户保持连接，消耗大量的运算资源，随着客户数目的增加性能不断下降，且存在死锁及崩溃的可能
- ❑ 数据模型“紧耦合”在每一客户端，数据库结构改变将导致全体客户端改变，不利于更新和维护
- ❑ 用户界面和程序逻辑绑在一起，代码难以重用
- ❑ 原始数据通过网络传递，加重网络负担
- ❑ 不适应在多用户、多数据库、且非安全的网络环境，可扩展性差
- ❑ 客户端应用程序越来越大，对使用者的要求也越来越高

◆3层/n层结构



- ❑ 客户端：业务数据模型与UI分开，更“瘦”
- ❑ 业务逻辑在中间层，通过连接池，多线程中间层可以处理多客户端
- ❑ 应用程序各部分之间松耦合，从而应用程序各部分的更新相互独立
- ❑ 业务逻辑集中放在服务器上由所有用户共享，使得系统的维护和更新变得简单，也更安全
- ❑ 数据库不再和每一个活动的用户保持一个连接，而是由应用程序组件负责与数据库打交道，降低数据库服务器的负担，提高了性能
- ❑ 组件之间和编程语言之间的变化不会影响互操作性



企业应用开发的需求

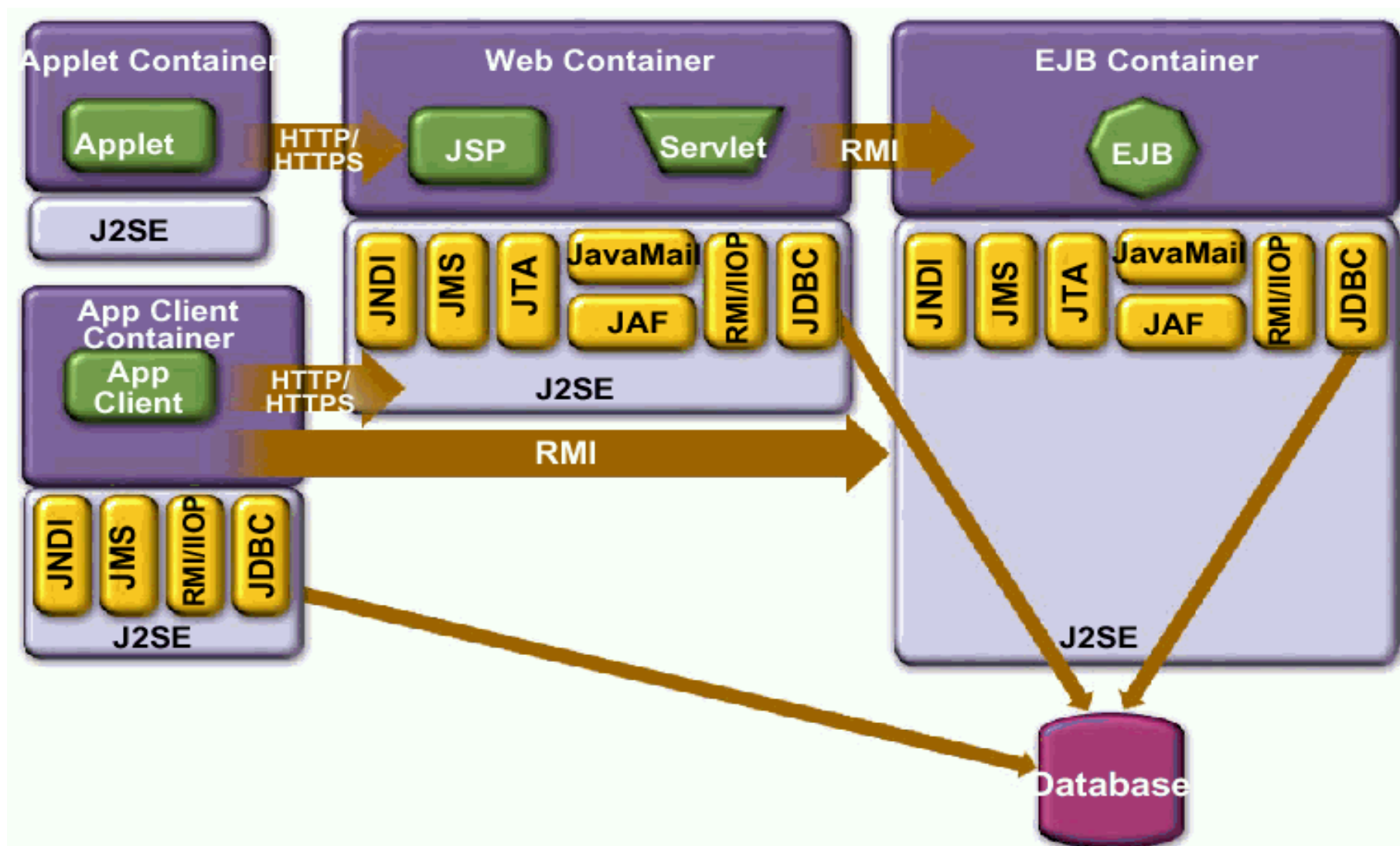
- ❑ 分布式
- ❑ 可移植
- ❑ 面向Web体系
- ❑ 可装配(螺丝刀)
- ❑ 能满足企业计算要求
 - ❖ 一致性
 - ❖ 事务性
 - ❖ 安全性
- ❑ 还要有好的特性
 - ❖ 可伸缩
 - ❖ 可扩展
 - ❖ 易维护

J2EE - Java 2 Enterprise Edition

- ❑ Open and standard based platform for developing, deploying and managing n-tier, Java-based, Web-enabled, server-centric enterprise applications
- ❑ 基于Java技术的，开放的、基于标准的平台，用以开发、部署和管理N层结构、面向Web的，以服务器为中心的企业级应用
- ❑ 主要内容
 - ❖ **J2EE规范**：开发商按照这个规范实现自己的J2EE系统。
 - ❖ **J2EE参考实现**：软件开发包（J2EE SDK）；J2EE的完全实现；一组工具和运行环境；完全免费得到。
 - ❖ **J2EE兼容测试包**：用来测试一种应用是否符合J2EE的规范。
 - ❖ **J2EE蓝图**：提供了用于实施基于J2EE的多层应用的文档和实例套件的编程模型，简化了开发基于J2EE的多层应用程序的复杂性。它应被用作开发人员设计和优化组件的原则，以便从策略上对开发工作进行分工，分配技术资源。



J2EE架构





什么是模式？

- ❑ 模式（pattern）：特定环境中问题的成功解决方案中的静态、动态结构，以及结构元素相互之间的协作关系。
- ❑ 模式其实就是解决某一类问题的方法论。把解决某类问题的方法总结归纳到理论高度，就是模式。
- ❑ 每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，你可以无数次地使用那些已有的解决方案，无需在重复相同的工作。



MVC设计模式

- MVC是目前广泛流行的软件设计模式；
- MVC正在成为在J2EE平台上推荐的一种设计模型；
- MVC是Model-View-Controller的缩写，即把一个应用的输入、处理、输出流程按照Model、View、Controller的方式进行分离，这样一个应用被分成三个层——模型层、视图层、控制层。
 - ❖ Model 数据层：应用系统的数据和业务逻辑的存放。
 - ❖ View 表示层：Model中的存储数据的可视化表示。
 - ❖ Control控制层：接受用户的输入，通知Model发生的事件。

- ❑ 视图代表用户交互界面，对于Web应用来说，可以概括为HTML界面，但有可能为XML和Applet等。
- ❑ 一个应用可能有很多不同的视图。
- ❑ MVC设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交给模型(Model)处理。
- ❑ 例如：一个订单视图只接受来自模型的数据并显示给用户，以及将用户界面的输入数据和请求传递给控制器和模型。

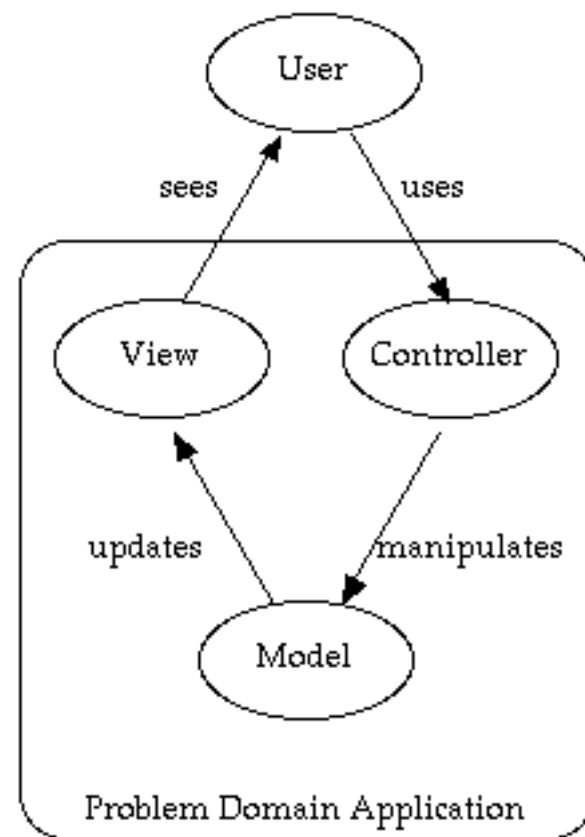
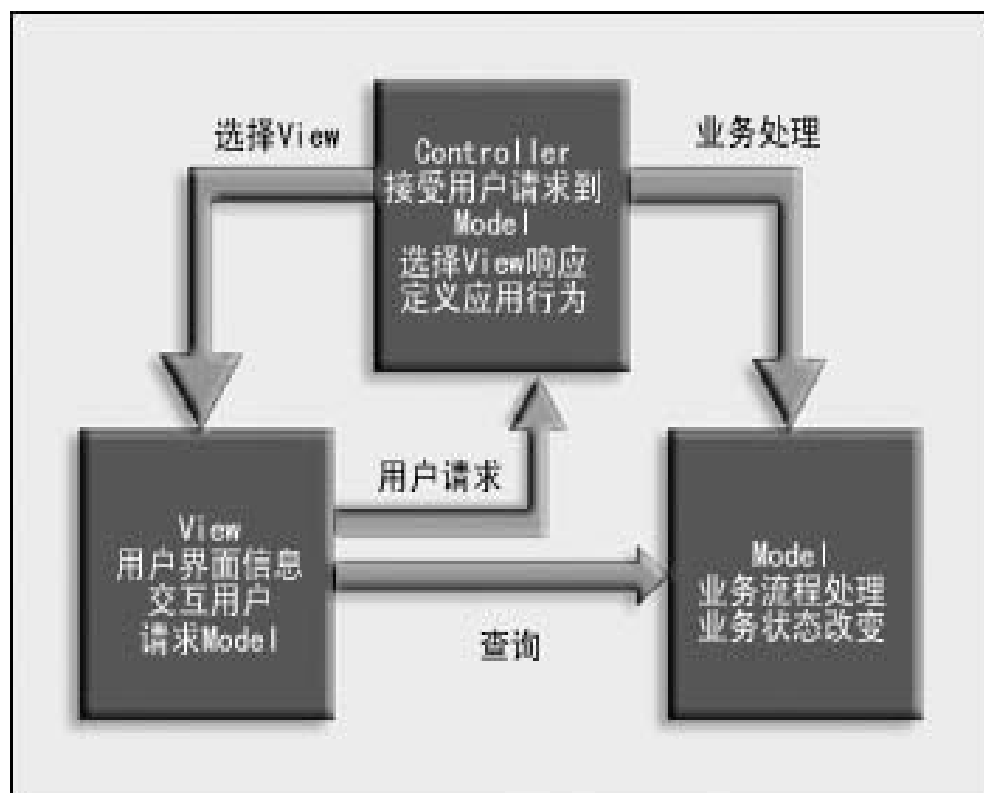
- ❑ 模型表示业务流程/状态的实际执行以及业务规则的维护（业务模型）。另外，还包括实体对象的数据保存（数据模型）。
- ❑ 业务流程的处理过程对其它层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。

- ❑ 可以理解为从用户接收请求, 将模型与视图匹配在一起, 共同完成用户的请求。
- ❑ 控制器相当于一个分发器, 它来决定选择什么样的模型和视图来完成特定的用户请求。
- ❑ 控制层并不做任何的数据处理。例如, 用户点击一个连接, 控制层接受请求后, 并不处理业务信息, 它只把用户的信息传递给模型, 告诉模型做什么, 选择符合要求的视图返回给用户。
- ❑ 一个模型可能对应多个视图, 一个视图可能对应多个模型。

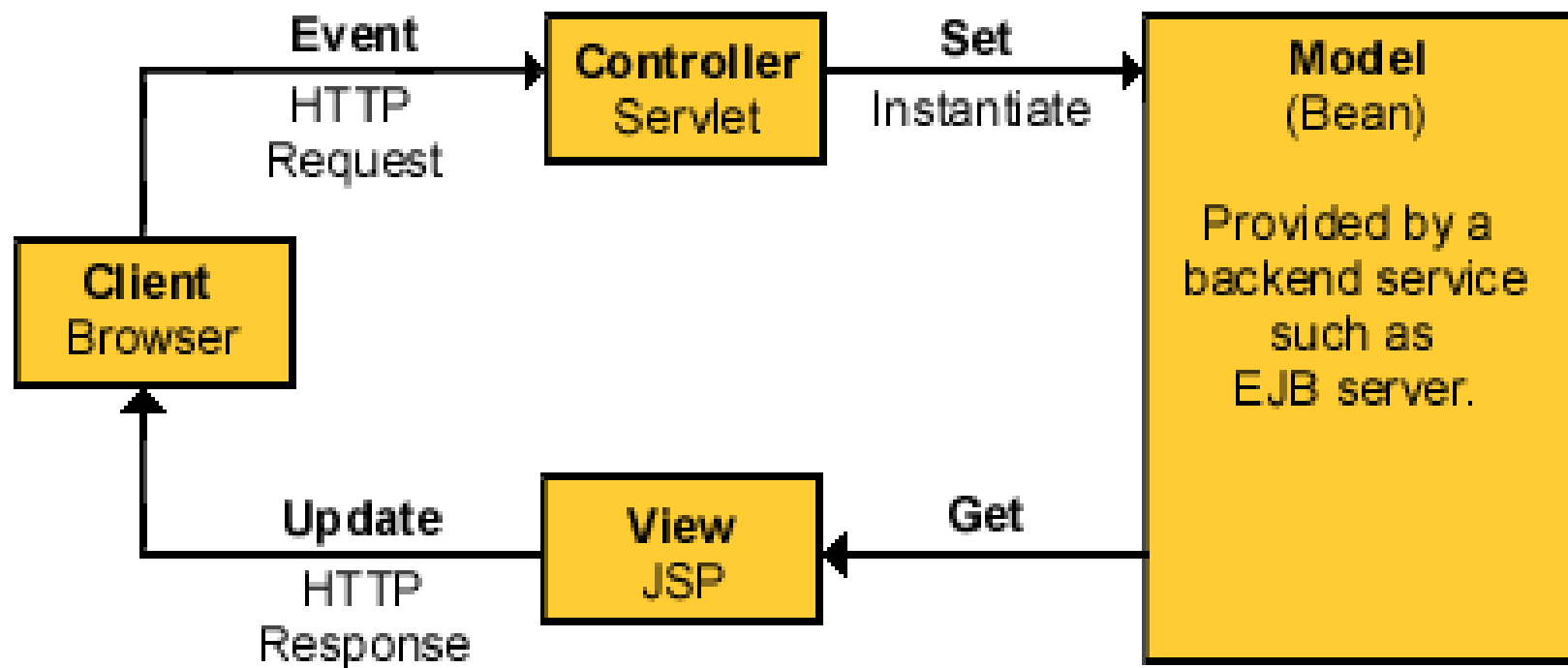
MVC设计模式的优点

- ❑ 一个模型可能对应多个视图。在目前用户需求的快速变化下，可能有多种方式访问应用的要求。在处理流程不变，但访问方式不同的情况下，按MVC设计模式，一个处理流程模型以及多个视图即可解决问题。这样减少了代码的复制，即减少了代码的维护量，一旦模型发生改变，也易于维护。
- ❑ 由于模型返回的数据不带任何显示格式，因而这些模型也可直接应用于接口的使用。
- ❑ 由于一个应用被分离为三层，因此有时改变其中的一层就能满足应用的改变。一个应用的业务流程或者业务规则的改变只需改动MVC的模型层。
- ❑ 控制层的概念也很有效，由于它把不同的模型和不同的视图组合在一起完成不同的请求，因此，控制层可以说是包含了用户请求权限的概念。
- ❑ 最后，它还有利于软件工程化管理。由于不同的层各司其职，每一层不同的应用具有某些相同的特征，有利于通过工程化、工具化产生管理程序代码。

MVC模式的工作流程



J2EE中MVC模式的对应实现



- 操作系统：Unix,Linux,Windows Server...,能够运行Java虚拟机即可。
- CERP系统运行在基于J2EE的应用服务器上。如：BEA WebLogic , IBM Websphere等。
- 客户端运行于Web浏览器中。
- 典型的Browser/Server结构。

- 客户端：IE浏览器
- 应用服务器：Weblogic Server 6.1
- 数据库服务器：Microsoft Sql Server 2000
- 开发工具：Borland Jbuilder

开发技术和语言：

- Java
- JSP
- XML
- Servlet
- Javascript

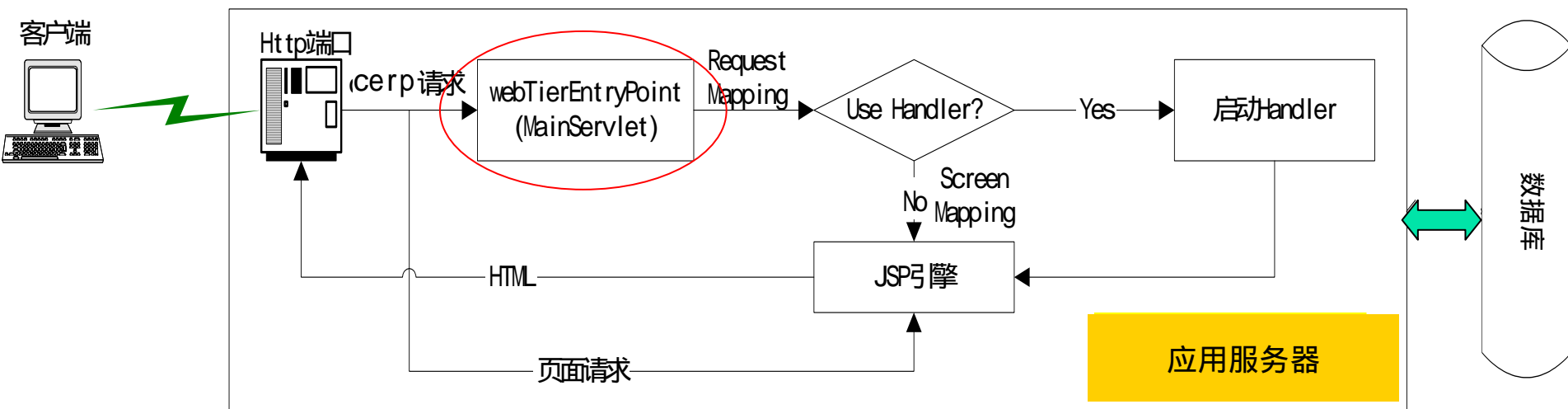
- 借鉴了SUN公司的Java BluePrints基础架构，通过改造其源代码得来。
- 使用了多种设计模式，具有良好的
 - ❖ 可维护性
 - ❖ 跨平台性
 - ❖ 扩展性
- 目前版本去掉了其中复杂的EJB应用部分，仅使用了其中的JSP+Servlet+JavaBean部分，简化了开发过程。

系统框架流程示意图



- ❑ JSP负责从用户接收请求，Servlet负责答复用户请求
- ❑ Servlet中处理请求中的数据、进行数据查询、以及相应的业务逻辑
- ❑ 处理结果转发给JSP页面进行格式化并显示

CERP系统框架运行流程(1)



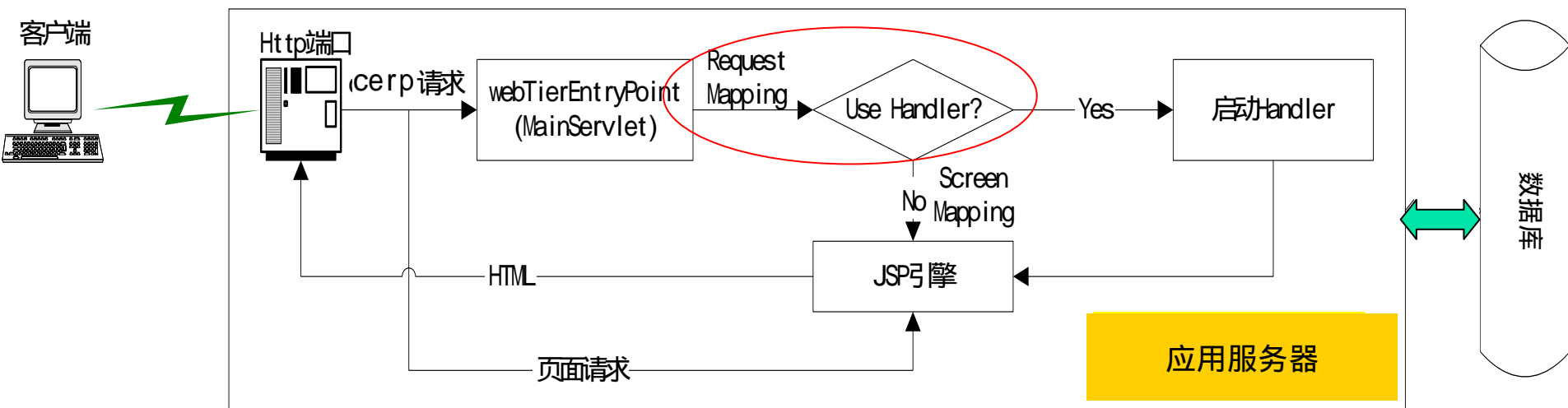
□ Step 1：中心控制

- ❖ 在cerp中，MainServlet实现controller的功能。
- ❖ MainServlet是cerp中控制系统的唯一入口，可在BEA Weblogic的web.xml配置文件中找到其定义。

```
<servlet>
  <servlet-name>webTierEntryPoint</servlet-name>
  <display-name>centralServlet</display-name>
  <description>no description</description>
  <servlet-class>com.huition.mainframe.control.web.MainServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>webTierEntryPoint</servlet-name>
  <url-pattern>/cerp/*</url-pattern>
</servlet-mapping>
```

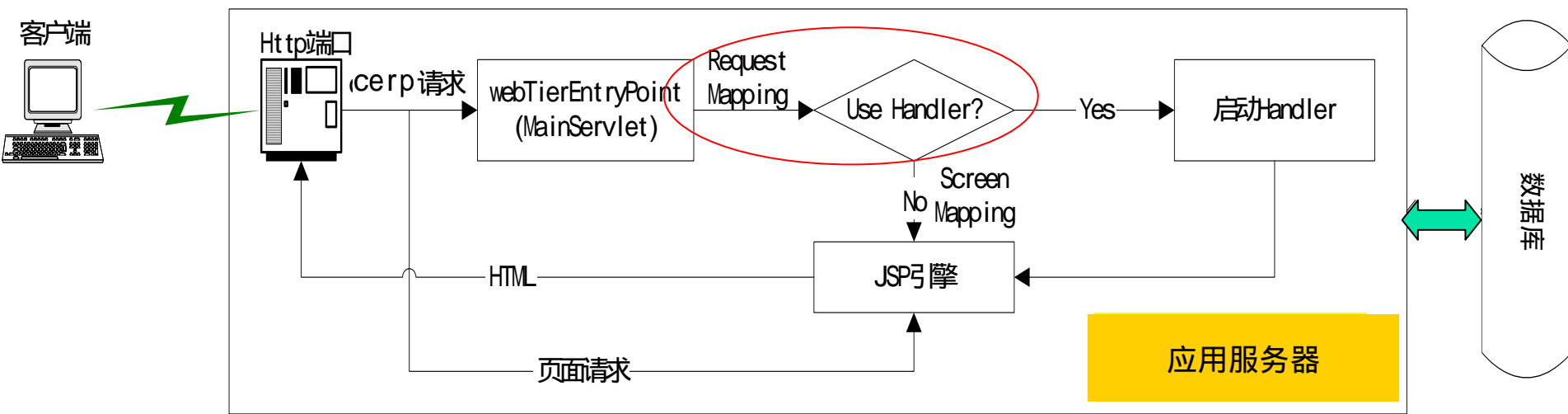
CERP系统框架运行流程(2)



□ Step 2：将http请求映射到action并执行之

- ❖ 首先从http请求中解析出的URL地址。
- ❖ 然后使用URL在request map中查找相应的action以及action对应的class，称之为Handler。
- ❖ 创建该handler的一个实例，并执行相关的操作，完成业务逻辑。
- ❖ Request map是一个xml配置文件，该文件中将每一个URL映射为一个特定的action。每个action对应着一个class，用来执行业务逻辑。
 - 在cerp中，request map由RequestMapping.xml文件实现。

CERP系统框架运行流程(2)



❑ 例如，当接收到

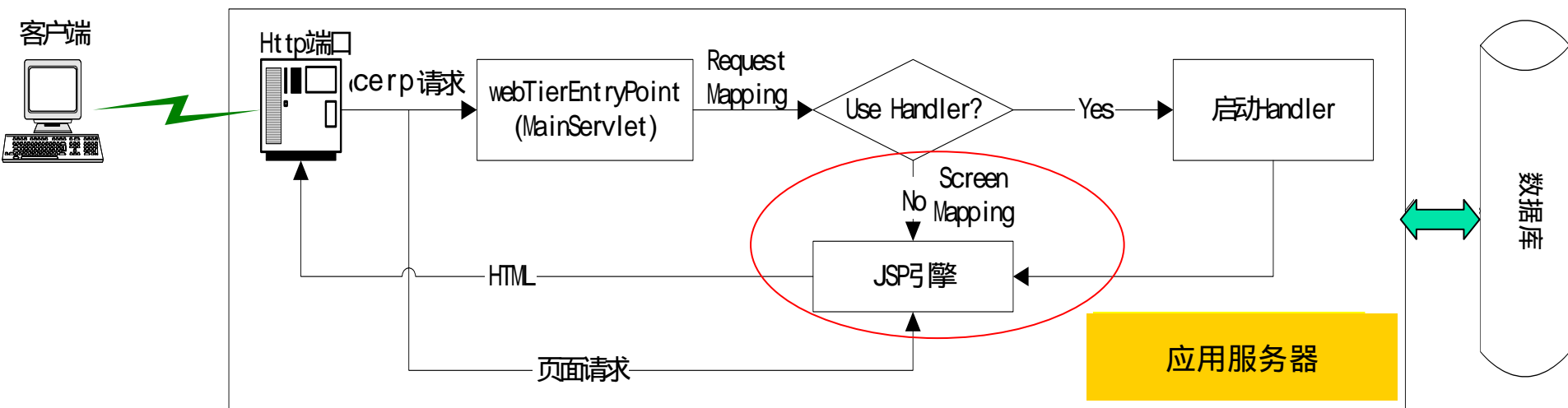
<http://localhost:7001/cerp/sam/sam2410>的http请求后，首先解析出的URI为
</sam/sam2410>

❑ 然后在RequestMapping.xml中可找到

```
<url-mapping url="/sam/sam2410" screen="SAM2410" useRequestHandler="true" requiresSignin="true">
    <request-handler-class>com.huiton.cerp.sam.SAM2410Handler</request-handler-class>
</url-mapping>
```

❑ 装载SAM2410Handler.class，并执行

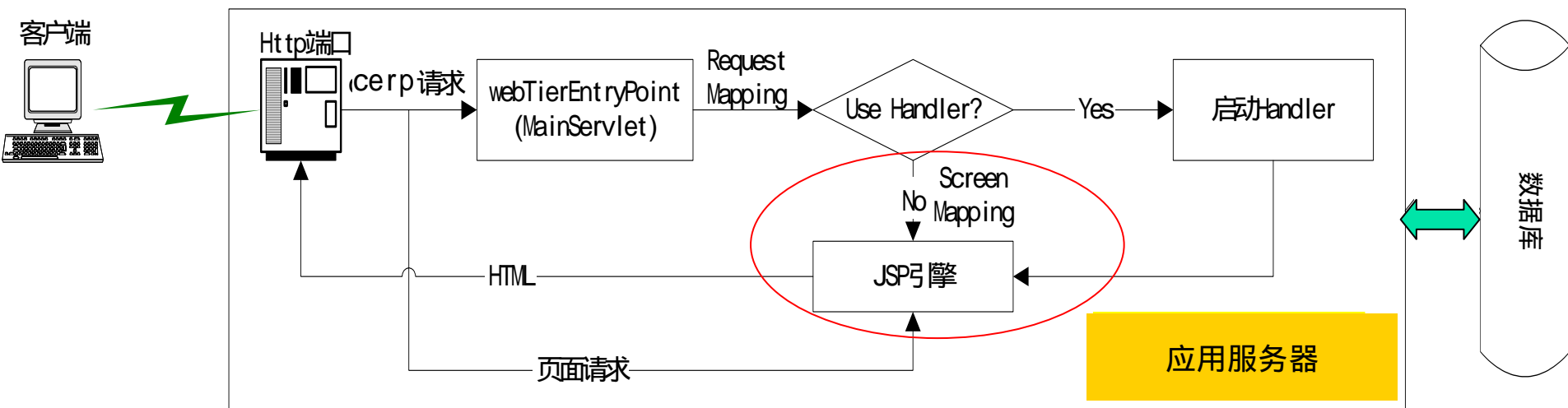
CERP系统框架运行流程(3)



❑ Step 3 : 选择view

- ❖ 当执行完business logic后，选择下一个需要显示给用户的界面。
- ❖ Screen flow map是一个xml配置文件，该文件中为每一个URL映射了下一步需要显示的界面（通常是JSP文件）。
 - 在cerp中 screen flow map 由ScreenDefinitions.xml中文件实现。

CERP系统框架运行流程(3)



❑ 例如，对 **/sam/sam2410** 的URL请求，RequestMapping.xml中相应的内容为：

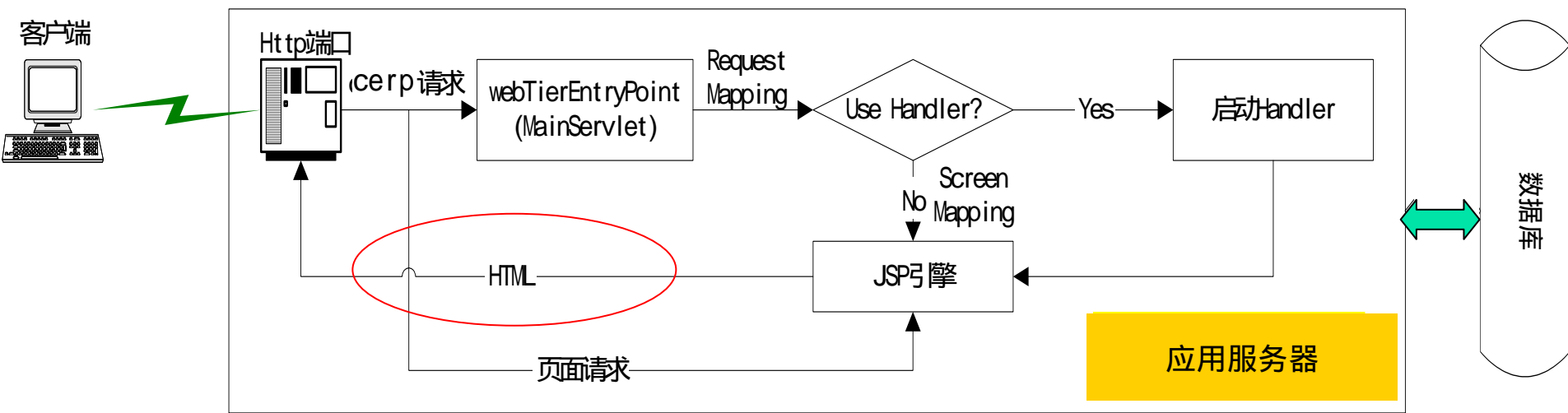
```
<url-mapping url="/sam/sam2410" screen="SAM2410" useRequestHandler="true" requiresSignin="true">
  <request-handler-class>com.huiton.cerp.sam.SAM2410Handler</request-handler-class>
</url-mapping>
```

❑ 在ScreenDefinitions.xml中，相应的映射为：

```
<screen>
  <screen-name>SAM2410</screen-name>
  <description></description>
  <parameter key="HtmlBanner" value="/com/huiton/cerp/pub/pages/PageStyle.jsp" direct="false"/>
  <parameter key="HtmlBody" value="/com/huiton/cerp/sam/sam2400/sam2410/sam2410.jsp"
    direct="false"/>
</screen>
```

2004年12月30日星期

CERP系统框架运行流程(4)

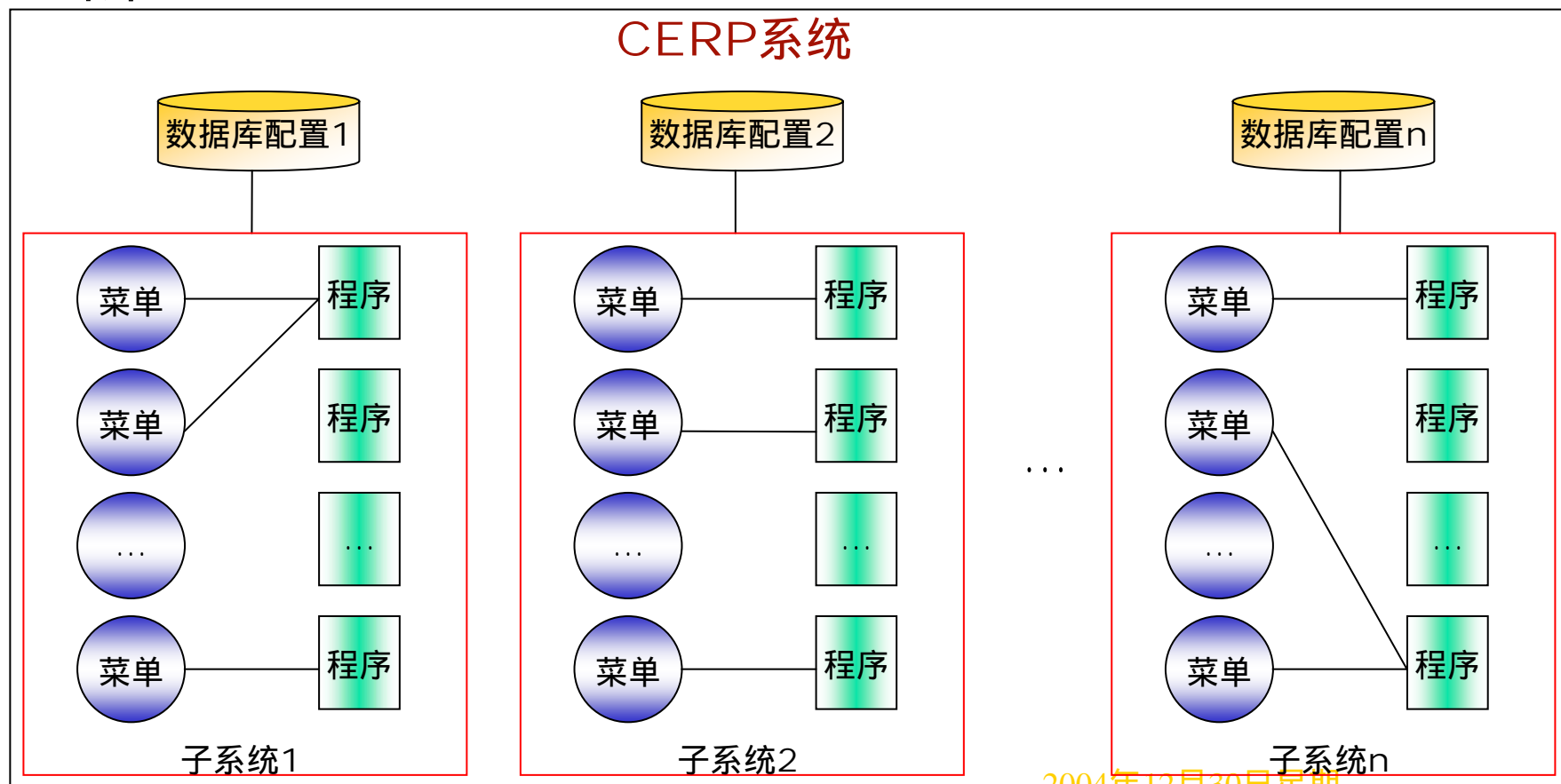


❑ Step 4 : 界面的组装

- ❖ 框架中预定义了具有CERP统一显示风格的界面模板，如界面的布局等。
- ❖ 组成界面的元素一般包括：
 - JSP
 - Tag
- ❖ 根据模板，将数据处理的结果显示到界面上。

CERP系统构成元素

- ❑ 子系统
- ❑ 数据库配置
- ❑ 程序
- ❑ 菜单



2004年12月30日 星期四



系统构成元素(1)

子系统

- ❖ 系统配置管理子系统SAM
- ❖ 离散生产管理子系统PPC
- ❖ 销售管理子系统 SAL
- ❖

子系统定义

▶ 查询 ▶ 重置

查询方式:



*代码	*名称	操作
CRM	客户关系管理	✕
MDM	制造数据管理系统	✕
PPC	离散生产系统	✕
SAL	销售管理	✕
SAM	系统管理	✕



1



第1页/共1页

2004年12月30日 星期三



系统构成元素(2)

数据库配置

- ❖ 数据库类型
- ❖ 数据库URL
- ❖ 数据库用户
- ❖ 密码

数据库配置

请修改

*子系统代码: PPC

*年份: 2004

*数据库代码: PPC

*数据库名称: 离散生产计划与控制

*操作系统: 10_windows

*数据库类型: 10_MSSQLSERVER

*驱动管理器: com.inet.tds.TdsDriver

*数据库URL: jdbc:inetdae:127.0.0.1:1433?database=newF:

*数据库用户: ppc

*数据库密码: ●●●●●●

*确认密码: ●●●●●●

▶保存返回 ▶保存 ▶重置 ▶返回

2004年12月30日星期

四



系统构成元素(3)

程序

- 一个子系统包含多个程序
- 程序属性：
 - ❖ 程序代码
 - ❖ 程序对应的URL
 - ❖ 程序名称

请修改

*子系统代码:	<input type="text" value="SAL"/>
*程序代码:	<input type="text" value="SAL1920"/>
*程序:	<input type="text" value="/cerp/SAL/SAL1920"/>
程序参数:	<input type="text"/>
属于:	<input type="text"/>
*程序名称(中):	<input type="text" value="产品基础价格手册"/>
程序名称(英):	<input type="text"/>
*程序级别:	<input type="text" value="0"/>
授权访问:	<input checked="" type="checkbox"/>
查询程序:	<input type="checkbox"/>
菜单结点:	<input type="checkbox"/>
结果集地址:	<input type="text"/>
表名:	<input type="text"/>

[▶保存返回](#) [▶保存](#) [▶重置](#) [▶返回](#)



系统构成元素(4)

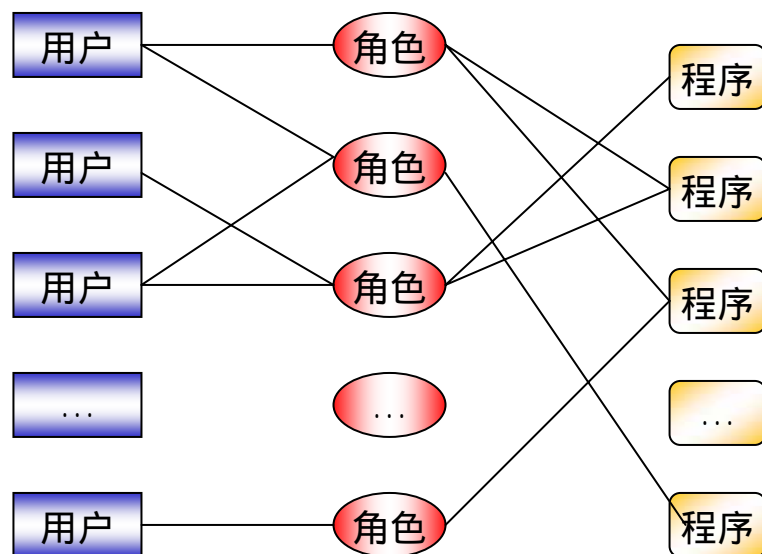
菜单

- ❑ 一个子系统对应多个菜单
- ❑ 一个菜单对应一个程序
- ❑ 菜单属性：
 - ❖ 菜单代码
 - ❖ 对应的程序代码
 - ❖ 菜单名称
 - ❖ 父菜单代码
 - ❖ 菜单顺序号

请修改

*子系统代码:	SAL	
*菜单代码:	ACR1150	
*程序代码:	ACR1150	
*菜单名称(中):	科目定义	
菜单名称(英):		
父菜单代码:	ACR_BASDAT	
菜单顺序号:	5	
叶子:	<input checked="" type="checkbox"/>	

▶保存返回 ▶保存 ▶重置 ▶返回



❑ 角色：具有访问程序的权限。

- ❖ 可通过“程序授权”功能决定该角色可以访问的程序。
- ❖ 只有经过授权的程序，才可以访问

❑ 用户

- ❖ 一个用户可以隶属于不同的角色，并同时拥有角色所对应的访问程序的权限。
- ❖ 用户具有唯一的用户名和密码，作为登录进入系统的标识。

系统权限控制：角色定义

角色定义

▶ 查询 ▶ 重置

查询方式:

角色定义

操作	角色代码	角色名称
 	admin	admin
 	epd	epd
 	ppc	ppc
 	sale	sale
 	sam	sam
 	scg	scg
 	wfs	wfs
 	<input type="text"/>	<input type="text"/>

1 第1页/共1页

角色代码: admin 角色名称: admin

▶ 查询 ▶ 重置

子系统代码 程序代码:

▶ 该用户拥有以下程序:

	子系统代码	程序代码	程序名称(中)	程序名称(英)
<input type="checkbox"/>	SAL	ACR1100	控制参数定义	Control Parameter Definition
<input type="checkbox"/>	SAL	ACR1110	付款条件定义	Payment Condition Definition
<input type="checkbox"/>	SAL	ACR1120	付款方式定义	Payment Style Definition
<input type="checkbox"/>	SAL	ACR1130	票据类型定义	Bill Type Definition
<input type="checkbox"/>	SAL	ACR1140	发票簿定义	Invoice Book Definition
<input type="checkbox"/>	SAL	ACR1150	科目定义	
<input type="checkbox"/>	SAL	ACR2100	应收单-定金结算	
<input type="checkbox"/>	SAL	ACR2110	退款处理	
<input type="checkbox"/>	SAL	ACR2120	应收账款调整	
<input type="checkbox"/>	SAL	ACR2130	预收账款调整	
<input type="checkbox"/>	SAL	ACR2140	应收单录入	ACR Bill Definition
<input type="checkbox"/>	SAL	ACR2150	退货/销售折让	Eject/Discount
<input type="checkbox"/>	SAL	ACR2160	应收款预警	ACR Prewarning
<input type="checkbox"/>	SAL	ACR2170	应收款催收单管理	Hasten Bill Management
<input type="checkbox"/>	SAL	ACR2180	预收定金录入	Prepay Management
<input type="checkbox"/>	SAL	ACR2190	客户回款录入	Customer Return Payment

1 第1页/共17页

☐ 当页全选 ☐ 全选 ▶ 删除 ▶ 增加 ▶ 返回

系统权限控制：用户定义

登录用户定义

▶查询 ▶重置

用户代码: _____ 用户名称: _____

操作	用户代码	用户名称
 	admin	admin
 	sale	销售部门
 	xhc	xhc
 		

1 第1页/共1页

请修改

用户唯一号: 338
用户名称: sale
用户名称: 销售部门
密码: ****

确认密码
用户角色
帐号状态
IP地址
允许数
密码可修改

▶保存返回 ▶保存

及更改角色





登录用户定义

	角色代码	角色名称
<input type="checkbox"/>	admin	admin
<input type="checkbox"/>	epd	epd
<input type="checkbox"/>	ppc	ppc
<input checked="" type="checkbox"/>	sale	sale
<input checked="" type="checkbox"/>	sam	sam
<input type="checkbox"/>	scg	scg
<input type="checkbox"/>	wfs	wfs

1 第1页/共1页

▶选取 ▶关闭

2004年12月30日 星期三



主要内容

❑ 系统开发框架介绍

- ❖ J2EE简介
- ❖ MVC设计模式
- ❖ CERP系统框架
 - 系统环境
 - 框架运行流程
 - 框架元素构成
 - 权限控制

❑ 软件开发流程

- ❖ 开发流程图
- ❖ 系统界面的设计与实现
 - 界面元素构成
 - 典型界面演示
 - 界面设计原则
 - 界面的开发
- ❖ 系统类的设计与实现
- ❖ 应用服务器的使用简介
- ❖ 实例演示

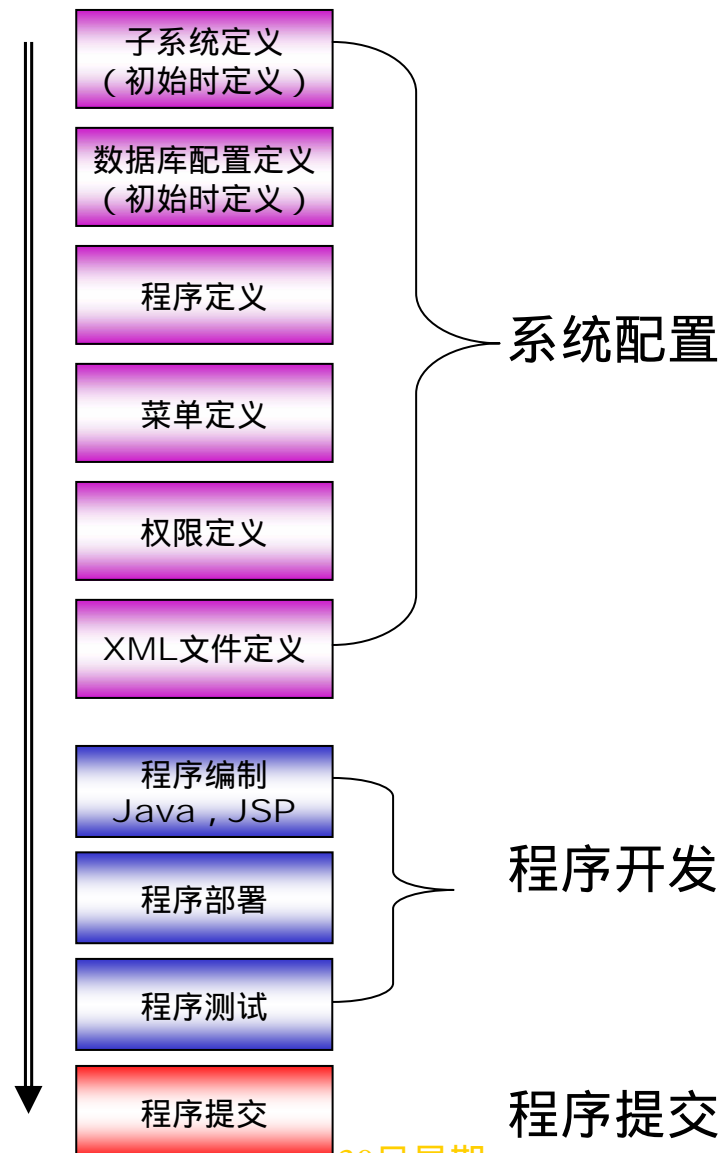
❑ 课后作业



系统程序开发流程

系统程序开发三个阶段

- **系统配置**: 完成程序、菜单、XML文件定义等工作，为代码的编制与测试准备好条件。
- **程序开发**: 编写handler和JSP代码，并部署到应用服务器中，进行调试和测试。
- **程序提交**: 提交测试完成的程序。



2004年12月30日 星期四

- 以程序作为开发过程中的基本单元
- 一个程序包含两个部分：
 - ❖ 用来执行业务逻辑的业务处理类(handler)，使用java语言书写，业务逻辑在此部分完成；
 - ❖ 用来显示处理结果的JSP页面，用JSP和JavaScript书写。



程序界面开发:界面元素构成

程序定义

查询 重置

子系统代码:

查询方式:

操作	子系统代码	程序代码	程序名称(中)	程序名称(英)
	MDM	mdm0000	空程序	
	MDM	MDM0101	物料维护	Material Defination
	MDM	MDM0102	物料分类码定义	Item Class Defination
	MDM	MDM0111	产品结构维护	BOM Defination
	MDM	MDM0112	替换物料维护	Substitute Defination
	MDM	MDM0113	BOM逐级展开	Expand BOM
	MDM	MDM0114	BOM综合展开	BOM Detail
	MDM	MDM0115	BOM逐级反查	Trace BOM
	MDM	MDM0121	工艺路线维护	Routing Defination
	MDM	MDM0122	工序工时维护	Operation Defination
	MDM	MDM0123	工作中心维护	Worker Center Defination
	MDM	MDM0124	加工设备维护	Machine Definaion
	MDM	MDM0125	工种维护	Work Type
	MDM	MDM0126	材料定额维护	Item Bation
	MDM	MDM0131	节日维护	Holiday Defination
	MDM	MDM0132	企业日历	Enterprise Calendar Defination

1 第1页/共14页

2004年12月30日星期

典型界面演示 (1) : 查询

CERP - Microsoft Internet Explorer

CERP 哈尔滨工业大学计算机学院
哈尔滨工业大学慧通新意信息技术有限公司 School of Computer Science and Technology Harbin Institute of Technology

欢迎 财务部 admin 登录 2004/02/18 16:14

主菜单

- 离散生产计划与控制系统
 - 主生产计划管理系统
 - 基础数据
 - 需求来源
 - 系统接口定义
 - 订单需求维护
 - 生产预测维护
 - 计划编制
 - 主生产计划编制
 - 主生产计划调整
 - 主生产计划审核
 - 主生产计划下达
 - 主生产计划撤销
 - 主生产计划结案
 - 信息查询
 - 主生产计划查询
 - 主生产计划汇总表
 - 计划来源跟踪反查
 - 可承诺订单数
 - 产前准备计划
 - 粗能力平衡系统
 - 基础数据
 - 粗能力平衡
 - 信息查询
 - 物料需求计划管理系统
 - 基础数据
 - MRP统计期定义
 - 计划号生成方式

主生产计划查询

物料代码: % 起止完工日期: - 工程号: MPS计划号: 计划状态: 全部

MPS计划号	工程号	物料代码	开工日期	完工日期	计划状态	制订人	制订日期
MPS03120008	3005	PKB63101	2003-12-30	2004-02-09	下达	admin	2003-12-21
MPS03120009	3007	PKA82101	2003-12-22	2003-12-30	下达	admin	2003-12-21
MPS03120010	3007	PKA82101	2003-12-22	2003-12-30	下达	admin	2003-12-21

1 第1页/共1页



典型界面演示（2）：多条信息录入修改界面

CERP - Microsoft Internet Explorer

CERP 哈尔滨工业大学计算机学院
哈尔滨工业大学慧通新意信息技术有限公司 School of Computer Science and Technology Harbin Institute of Technology

欢迎 财务部 admin 登录 2004/02/18 16:14

主菜单

- 入库通知单
- 信息查询
 - 生产自制需求查询
 - 采购需求查询
 - 生产自制需求分析
 - MRP计划汇总查询
- 能力需求计划系统
- 辅助模块
- 车间任务管理子系统
- 车间作业管理子系统
- 车间数据管理子系统
- 车间生产统计子系统
- 系统配置管理
 - 软件配置
 - 数据库类型查询
 - 数据库驱动程序查询
 - 操作系统类型查询
 - 数据库配置
 - 菜单管理
 - 子系统定义**
 - 程序定义
 - 菜单定义
 - 权限管理
 - 公司定义
 - 部门定义
 - 职位定义
 - 角色定义
 - 登录用户定义

子系统定义

查询方式:

*代码	*名称	操作
CRM	客户关系管理	<input type="button" value="X"/> <input type="button" value="编辑"/>
MDM	制造数据管理系统	<input type="button" value="X"/> <input type="button" value="编辑"/>
PPC	离散生产系统	<input type="button" value="X"/> <input type="button" value="编辑"/>
SAL	销售管理	<input type="button" value="X"/> <input type="button" value="编辑"/>
SAM	系统管理	<input type="button" value="X"/> <input type="button" value="编辑"/>
		<input type="button" value="新增"/> <input type="button" value="删除"/>

1 第1页/共1页

完毕 Internet



典型界面演示 (3) : 单条信息录入修改界面

CERP - Microsoft Internet Explorer

CERP 哈尔滨工业大学计算机学院
哈尔滨工业大学慧通新意信息技术有限公司 School of Computer Science and Technology Harbin Institute of Technology

欢迎 财务部 admin 登录 2004/02/18 16:14

主菜单

- 离散生产计划与控制系统
 - 主生产计划管理系统
 - 基础数据
 - 需求来源
 - 系统接口定义
 - 订单需求维护
 - 生产预测维护
 - 计划编制
 - 主生产计划编制
 - 主生产计划调整
 - 主生产计划审核
 - 主生产计划下达
 - 主生产计划撤销
 - 主生产计划结案
 - 信息查询
 - 主生产计划查询
 - 主生产计划汇总表
 - 计划来源跟踪反馈
 - 可承诺订单数
 - 产前准备计划
 - 粗能力平衡系统
 - 基础数据
 - 粗能力平衡
 - 信息查询
 - 物料需求计划管理系统
 - 基础数据
 - MRP统计期定义
 - 计划号生成方式

订单需求维护

请修改

*合同号: Order_35/1

*工程号: 3004 确认标志: ☒

*物料编码: PKA82103

物料名称: P6×2-8-111-1.6S-23.4/L-I a

规格型号:

位号:

订单状态: 未处理

外协标志: ☐ 外协单位:

*订货数量: 2.0

*出厂日期: 2004-01-08 *交货日期: 2004-01-10

*产值(万元): 5.0 *产量(吨): 5.0

订货单位:

备注:

填写人: admin

填写日期: 2003-12-21

保存 重置 返回 打印预览 打印

完毕 Internet



典型界面演示（4）：单据录入界面

CERP - Microsoft Internet Explorer

哈工大计算机学院

材料入库通知单查询

材料入库通知单

*第ADMI20030800011号

*制定日期: 2003-08-12

*质量检验人: 迟世军	*制定人: 暴锡亭	币种: 人民币(元)
*物料验收单据号: YX2220030800013	订单号: PR0320030800020	发票号:
*接受仓库: 小金库房	*采购员姓名: 暴锡亭	估价标记: 估计价格
物料验收入库时间: 2003-08-12	发货单位代码: 00548	发货单位名称: 哈尔滨汇工工具有限公司
本单位: 哈尔滨空调股份有限公司	*实际总金额: 0.00	*采购处审核时间: 2003-11-08
采购审核人姓名: 管理员		
*采购处审核意见: 采购处通过	采购处审核说明:	返回

▶打印预览 ▶打印

序号	*物料编码	物料名称	计量单位	物料规格	实收数量	*增值
1	W603MHGB1438.1/25	锥柄麻花钻	支	Φ25::GB1438.1:	-30.0000	17.00
2	W603KKGB1141/25	扩孔钻	支	Φ25::GB1141:	30.0000	17.00

1 第1页/共1页

完成 Internet

开始 快捷方式 st... SQL Server ... JBuilder 8 - ... SQL 查询分... CERP - Micro... CERP - Micro... CERP - Micro... 16:41



典型界面演示 (5) : 分类显示处理信息界面

CERP - Microsoft Internet Explorer

华润酒精
CR Alcohol

ERP

诚信 务实 专业 团队 积极 创新
哈尔滨工业大学慧通新意

欢迎 admin 登录2004/02/18 16:37

主菜单

成本管理

计量管理系统

基础数据管理

生产管理系统

- 生产定义文件
 - 车间定义
 - 生产线定义
 - 管理控制点定义
 - 岗位定义
 - 生产人员定义
 - 工艺参数类型定义
 - 工艺参数定义
 - 班组班别定义
- 主配方管理
- 生产计划管理
- 生产数据管理
- 能耗计划管理
- 能耗数据管理
- 劳动管理
- 生产调度命令管理
- 物料管理
- 生产统计管理

采购管理系统

库存管理系统

设备管理系统

质量管理系统

销售管理系统

综合查询系统

生产线定义

生产线基本信息

部门名称:	生产技术部	车间名称:	预处理车间	生产线代码:	050402
生产线名称:	粮储供料生产线	管控点数量:	1	设备数量:	0
岗位数量:	21	生产线主任:		生产线电话:	
说明:	预处理车间粮储工段第二				

▶管控点定义 ▶设备定义 ▶岗位定义 ▶班组定义

操作	班组代码	班组名称	班组人数	组长	说明
✖	050402A	甲班	24	魏立君	
✖	050402B	乙班	24	朱 平	
✖	050402C	丙班	23	关成达	
✖	050402D	丁班	23	陈忠华	

1第1页/共1页

▶返回

2004年12月30日 星期四

本地 Intranet



界面设计原则(1)

- ❑ 能够充分表达设计思想和需求信息。
- ❑ 界面要求直观明了，用户接触软件后对界面上对应的功能应该一目了然、不需要多少培训就可以方便使用。
- ❑ 用户界面应当按照用户的思想来控制应用如何工作、如何响应，而不是由开发者按自己的意愿把操作流程强加给用户。
- ❑ 应保持各个界面的一致性
 - ❖ 使用标准的控件；
 - ❖ 使用相同的信息表现方法，如在字体、标签风格、颜色、术语、显示错误信息等方面确保一致。
 - ❖ 标准的操作方法。

界面设计原则(2):字体

- ❑ 标签、题头：黑色宋体，5号字，加粗
- ❑ 数据内容：黑色宋体，5号字
- ❑ 操作按钮：红色宋体、5号字

- ❑ 不得擅自使用其他字体、字号和颜色

界面设计原则(3):分辨率

- ❑ 按照800*600的屏幕分辨率编写界面；
- ❑ 表格的最大宽度不超过760像素
- ❑ 如果数据量太多，一行内无法显示完全，使用滚动条。
- ❑ 如果一页显示多行，则要加上分页操作。

界面设计原则(4):布局

- ❑ 操作顺序从上到下、从左到右
- ❑ 首先应输入的和重要信息应当靠前，位置也应放在窗口上较醒目的位置
- ❑ 布局力求简洁、有序、易于操作

界面设计原则(5):出错信息和警告

- ❑ 信息以用户可以理解的术语描述；
- ❑ 信息应提供如何从错误中恢复的建设性意见；
- ❑ 信息应指出错误可能导致那些不良后果，以使用户检查是否出现了这些情况或帮助用户进行改正；
- ❑ 信息不能带有判断色彩，即任何情况下不能指责用户（不得使用感叹号）
- ❑ 必须使用中文进行提示
- ❑ 程序调试过程中加入的对话框，在调试通过后必须去掉。

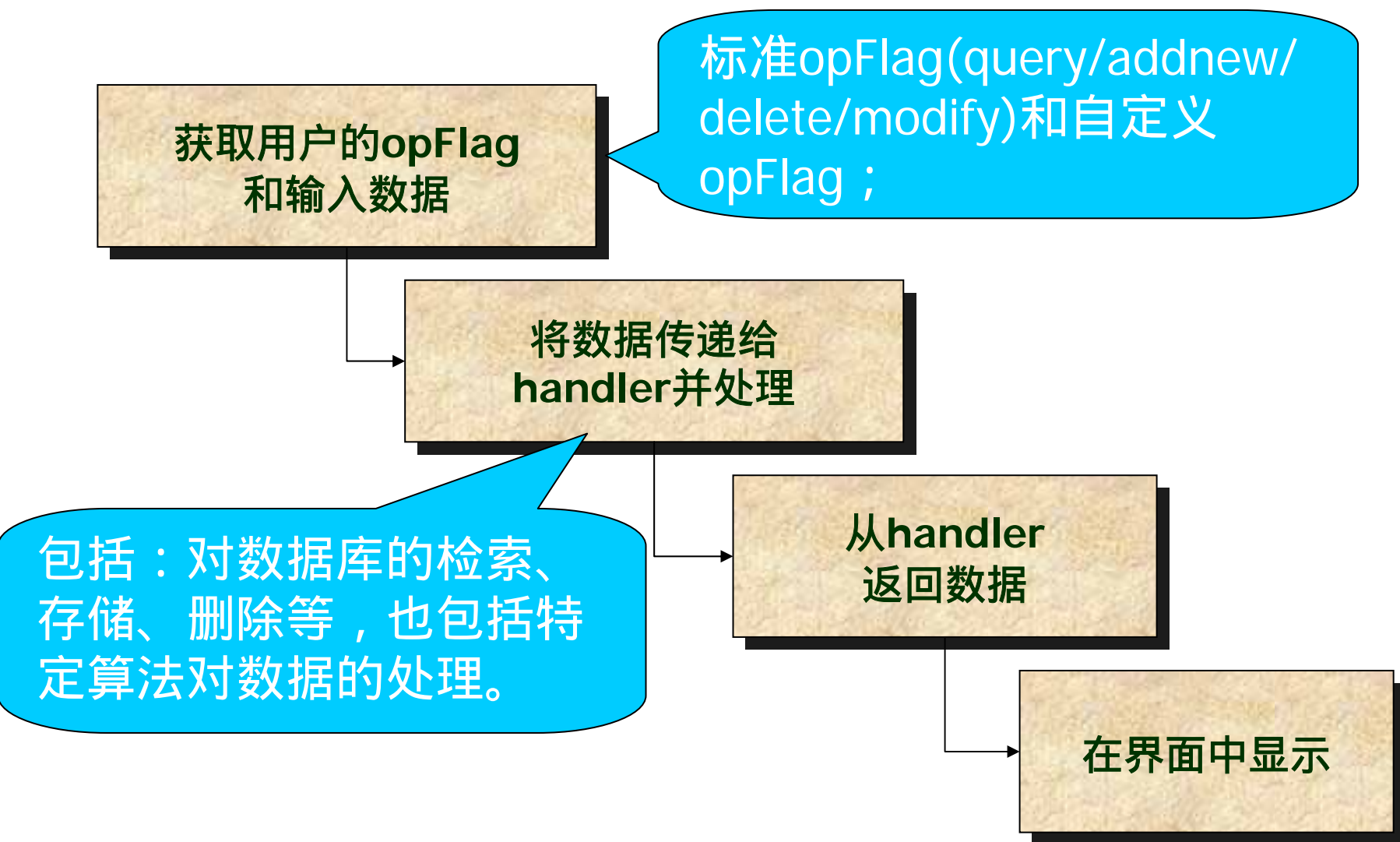
界面设计原则(5):界面的划分

- 界面上的所有操作都是由企业的同一个角色完成的；如果不同的操作由不同的人完成，则应该分成两个界面。

界面设计原则(6):查询条件的处理

- ❑ 每一个查询界面都要给出查询条件;
- ❑ 如果用户没有输入查询条件而执行了查询操作,则不应显示任何结果信息;
- ❑ 查询条件不宜过多(重点不突出,增加编程负担);也不宜过少(满足不了用户需求);
- ❑ 查询条件的设定应以所查数据的要点(关键项)和用户的主要查询习惯为准。例如:查询物料主文件时,用户比较熟悉的是物料名称、规格型号等项,则这些项要做为查询条件。
- ❑ 如果某信息,既有编码,又有相应的名称,则以名称为准,也可二者均采用,不要只查询编码,忽略名称。
- ❑ 编程中尽量使用模糊查询,增大查询范围。

界面的开发:界面运行流程



2004年12月30日星期

- 对于单一数据，以字符串变量的形式传递；
 - ❖ `String name=(String)request.getAttribute("name");`

- 对于二维数据，以向量数组的形式传
 - ❖ `Vector vct = (Vector) request.getAttribute("vct");`

- ❑ 表示用户在该界面上所采取的动作，是控制操作和页面调用的标志；
- ❑ CERP的标准界面通常具有浏览、添加、修改等通用功能，另外还包含一些自定义的功能，如审核、过账等业务操作；
- ❑ 有些操作在后台运行，无需用户参与，如删除、保存等；
- ❑ 有些操作则需要用户的参与，如选择、输入查询条件、输入必要数据等。
- ❑ 针对每一项操作，都定义了唯一的opFlag与之对应,如：“query”代表查询，“delete”代表删除，“modify”代表修改，“addnew”代表新增等。
- ❑ opFlag值的修改，一般通过Javascript函数处理。

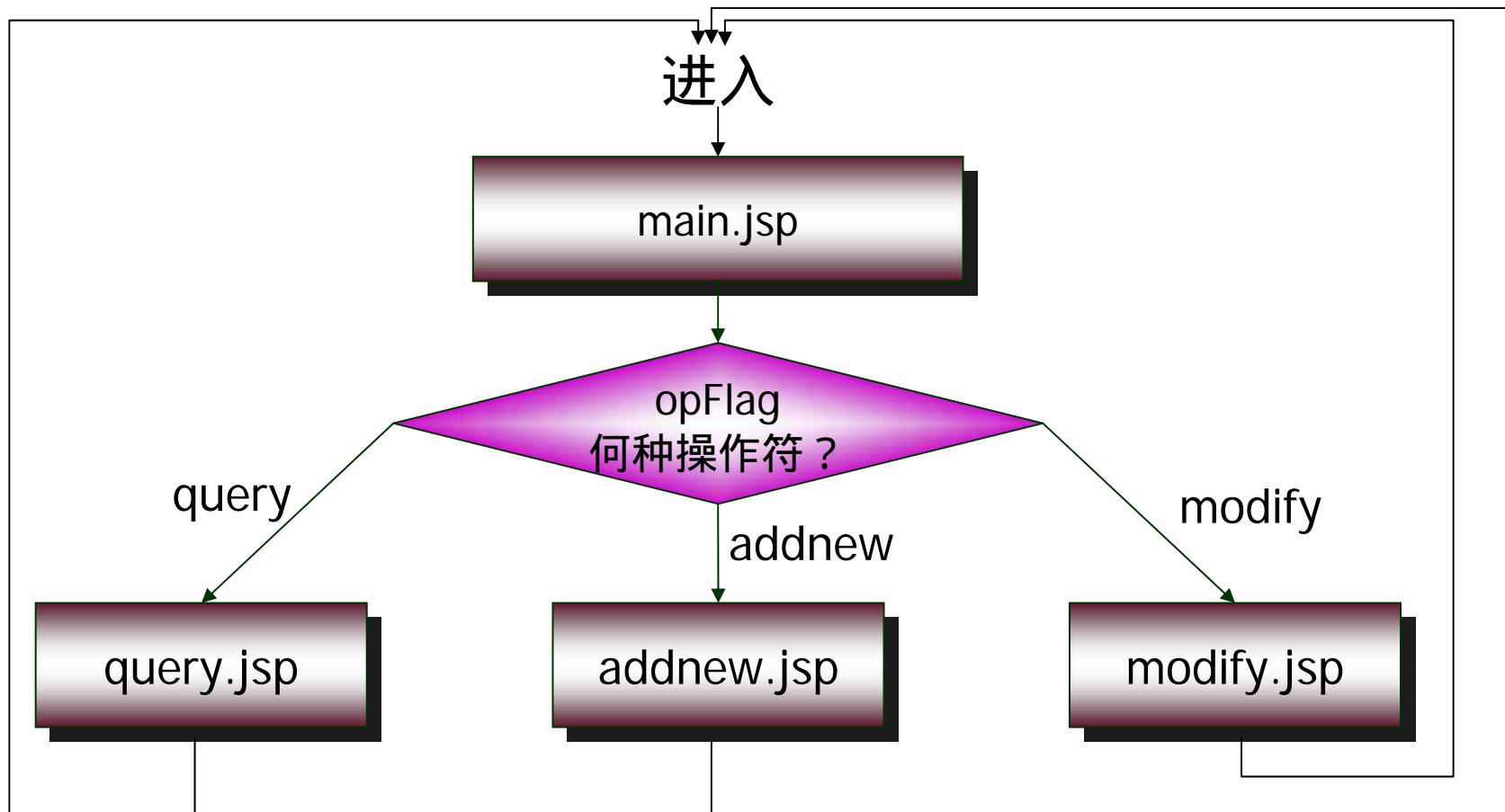
错误判断标志

用来从handler向JSP传递是否存在错误，JSP根据outFlag的值来判断是否需要提示。

页面中需要的JSP文件

- ❑ 主JSP文件：此文件一般不处理显示，用来进行页面调用控制，根据opFlag决定调用哪个子界面；
- ❑ 对标准界面中的需要单独显示的子界面，要编写相应的JSP文件；
- ❑ 例如，一个包含查询、新增、删除、修改功能的标准界面所需的JSP文件包括：
 - ❖ XXX_main.jsp
 - ❖ XXX_query.jsp
 - ❖ XXX_addnew.jsp
 - ❖ XXX_modify.jsp

JSP文件的调用顺序





Handler的开发

- ❑ 一个程序对应一个handler
- ❑ 在requestmappings.xml中通过程序的URL与handler关联起来。
- ❑ 一个handler的基本构成：
 - ❖ 获取从客户端界面提交的数据
 - ❖ 列出所有可能的opFlag，对opFlag的每一种可能的值，编写业务逻辑
 - ❖ 处理完成后得到的结果传递给界面并显示



系统公共类

- 包括
 - ❖ 数据库访问操作类
 - ❖ 语言工具类
 - ❖ 加密工具类
 - ❖ 字符串处理类
 - ❖ 日期处理类
 - ❖ 等等
- 可以在自己的程序中直接使用这些公共类
- 具体的使用方式参阅相应文档。
- 数据库查询类：com.huiton.pub.dbx.pageQuery



开发流程总结

- ❑ 在 requestmappings.xml 文件中定义
 - ❖ 程序对应的URL 路径与显示页面的名称的映射
 - ❖ 请求处理类(handler)的名称
- ❑ 在screendefinitions.xml文件中定义
 - ❖ 页面名称与JSP的对应关系
- ❑ 编写JSP文件
- ❑ 编写请求处理类(handler)
- ❑ 部署到应用服务器
- ❑ 编译执行

Weblogic的使用 (1)

- 服务器的启动：

x:\bea\wlserver61\config\mydomain\startWebLogic.cmd

- **handler**存放位置：

x:\bea\wlserver61\config\mydomain\applications\DefaultWebApp\WEB-INF\classes

- **JSP**程序存放位置：

x:\bea\wlserver61\config\mydomain\applications\DefaultWebApp

- **XML**文件存放位置：

x:\bea\wlserver61\config\mydomain\applications\DefaultWebApp\WEB-INF\xml

■客户端如何调用程序

在浏览器的地址栏中输入如下地址：

<http://服务器的ip地址:7001/访问程序的地址>

如：

<http://202.118.239.177:7001/cerp/signin>

■调试程序的注意事项

- JSP程序在发生变动之后，不需要重新启动应用服务器；
- handler程序发生变动之后，需要重新启动应用服务器
- XML文件发生变动之后，需要重新启动应用服务器

演示一个简单的增删改查询功能

BOM表的查询

- ❑ 阅读先前已经实现的代码，找到相应的功能。
- ❑ 查阅资料（各种技术文档）
- ❑ 咨询同事，互相切磋
- ❑ 请教每组的负责人
- ❑ 联系
 - ❖ 李成严：chengyan@hit.edu.cn
 - ❖ 王忠杰：rainy@hit.edu.cn
 - ❖ 聂兰顺：rhine@hit.edu.cn
 - ❖ 乔立民：qlm@hit.edu.cn
 - ❖ 徐汉川：xhc@hit.edu.cn

祝大家开发顺利!
多谢!