IBM Software Group

SOA 开发第三步 – 构建服务组件

IBM Rational　姚炳雄 yaobx@cn.ibm.com

e business on demand.

SOA 开发三步曲

第一步 业务分析员

业务目标
业务模型
业务需求

- 分析并记录现有的业务流程及其概念
- 优化改进业务流程

Rational. software
Rational Requisite Pro
Rational Software Modeler

第二步 架构设计师

服务模型
软件架构
企业架构

- 确定服务
- 构建服务模型
- 搭建系统架构

Rational. software
Rational Software Architect

集成开发员

服务流程模型
服务组装模型

- 实现业务流程和组合应用
- 定义服务

WebSphere software
Websphere Integration Developer

第三步 开发人员

设计模型
实现模型

- 设计服务组件
- 编码实现服务

Rational. software
Rational Application Developer

资源组合
共享资产
公共流程

Rational. software
Rational Portfolio Manager
Rational Unified Process
Rational ClearCase
Rational ClearQuest

内容

- 快速构建Web Service
- 快速构建SDO
- 快速构建JSF

构建服务组件 - SOA开发第三步

- 快速构建Web Service
- 快速构建SDO
- 快速构建JSF

快速构建Web Service－ Web Service简介

SOAP

- 交换信息的协议
- 多平台支持的xml协议，用于不同平台 的client / Server之间通信，

WSDL - Web Services Description Language

- 基于xml的描述的服务接口.

UDDI - Universal Description, Discovery, Integration

- UDDI 服务提供的列表信息
- 使用 SOAP 应用 更新或查询服务列表信息，以取得要访问的服务信息

快速构建Web Service－Web Service简介

JAX-RPC

- 基于XML 的RPC 的API
- 从Servlet 中访问WebService的API
- 客户端访问 Web Services的相关API
- 如何将JAVA 类型映射成WSDL中的XML

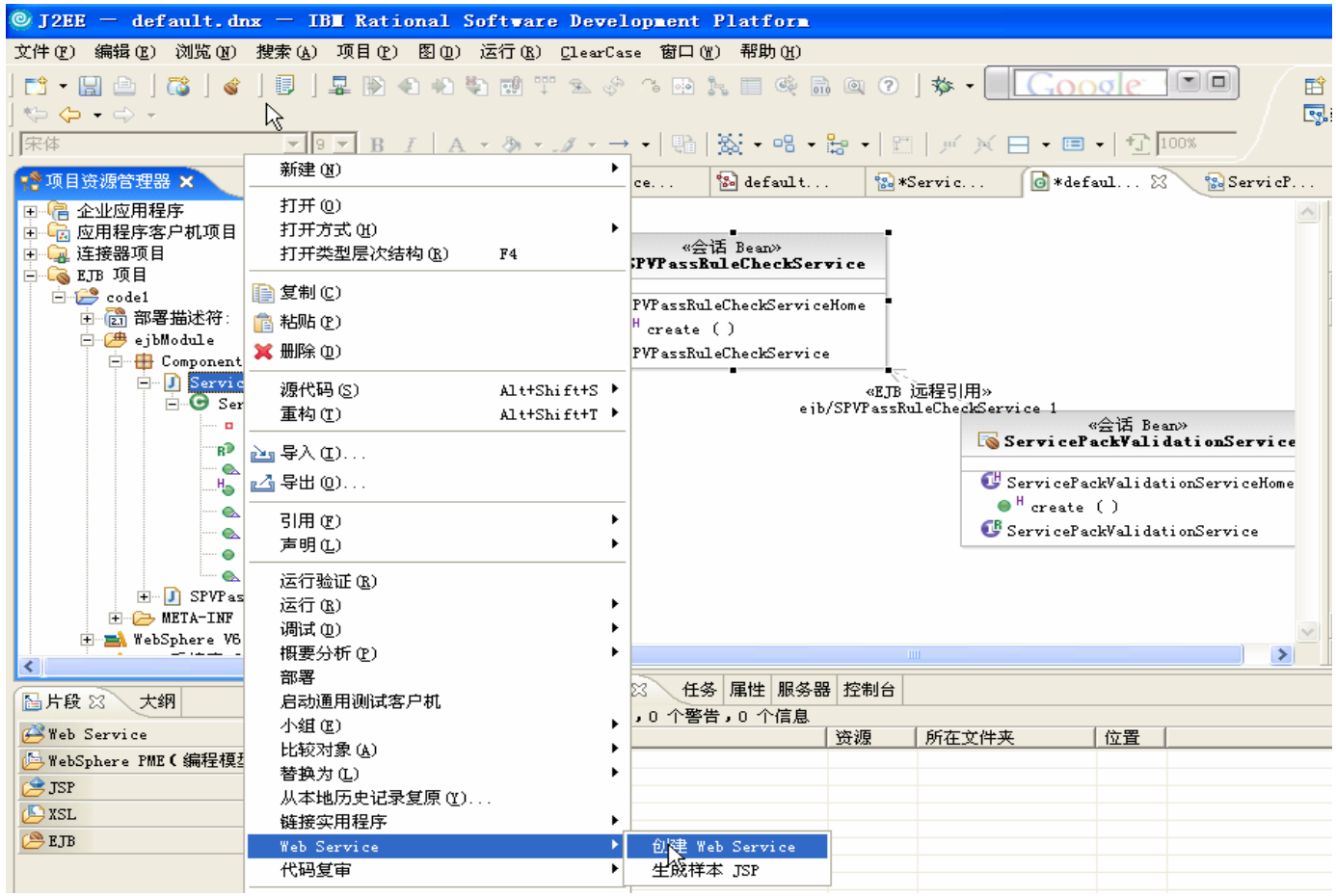
JSR-109

- 如何将J2EE应用打包，部署成WebService.
- 无状态Session Bean如何转换到WebService.

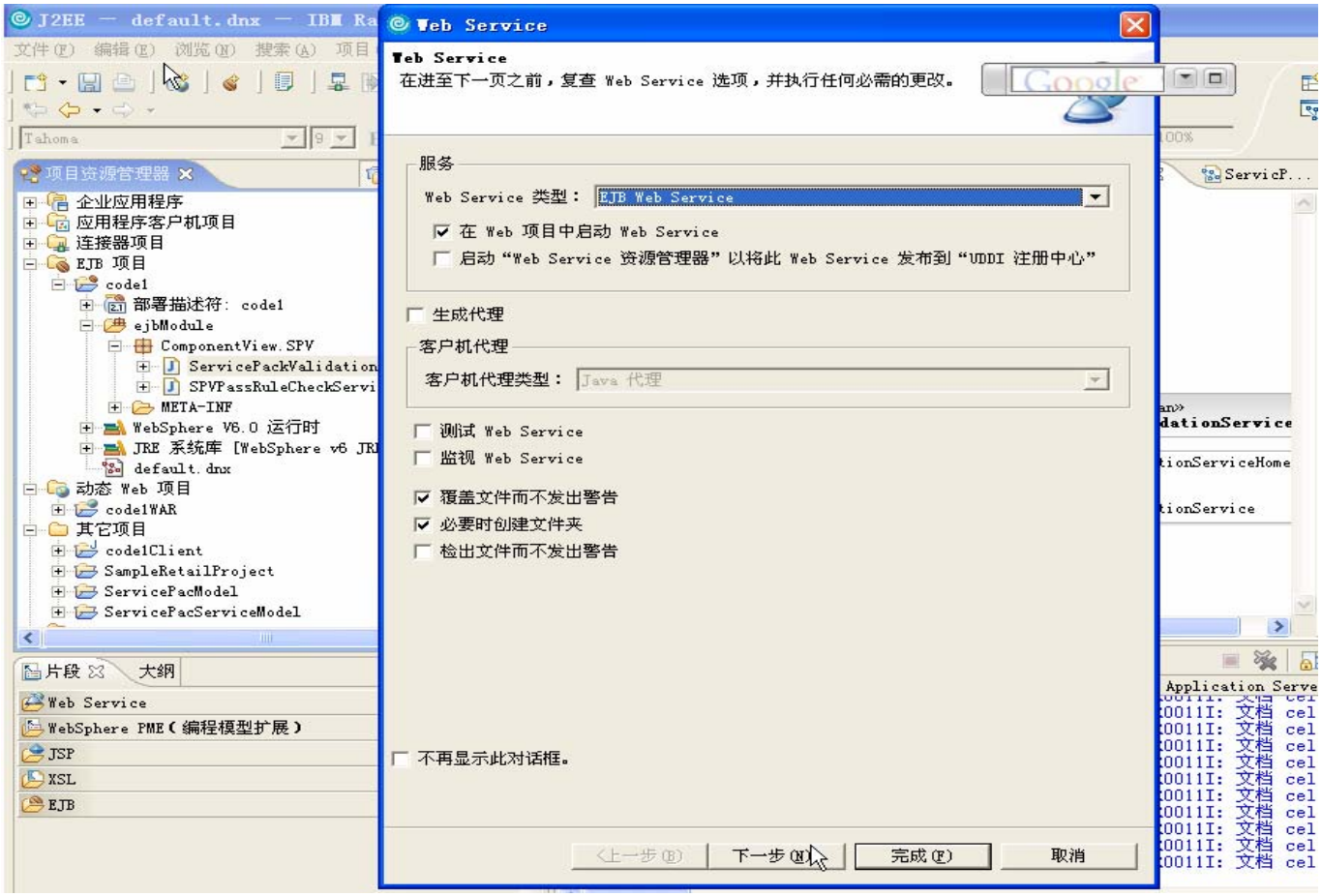
快速构建Web Service－RAD的开发能力

- 开发
  - Bottom Up – 将已经存在的 JavaBeans, EJBs, DB2进行Web Service封装
  - Top Down – 从WSDL自动生成JAVA调用的桩代码
- 测试
  - 使用WSDL explorer 进行Web Service执行测试
- 部署
  - 将Web Service 部署到WebSphere® 应用服务器 或者 Tomcat 环境
- 发布
  - 将Web services 发布到 UDDI v2 or v3 注册服务器上

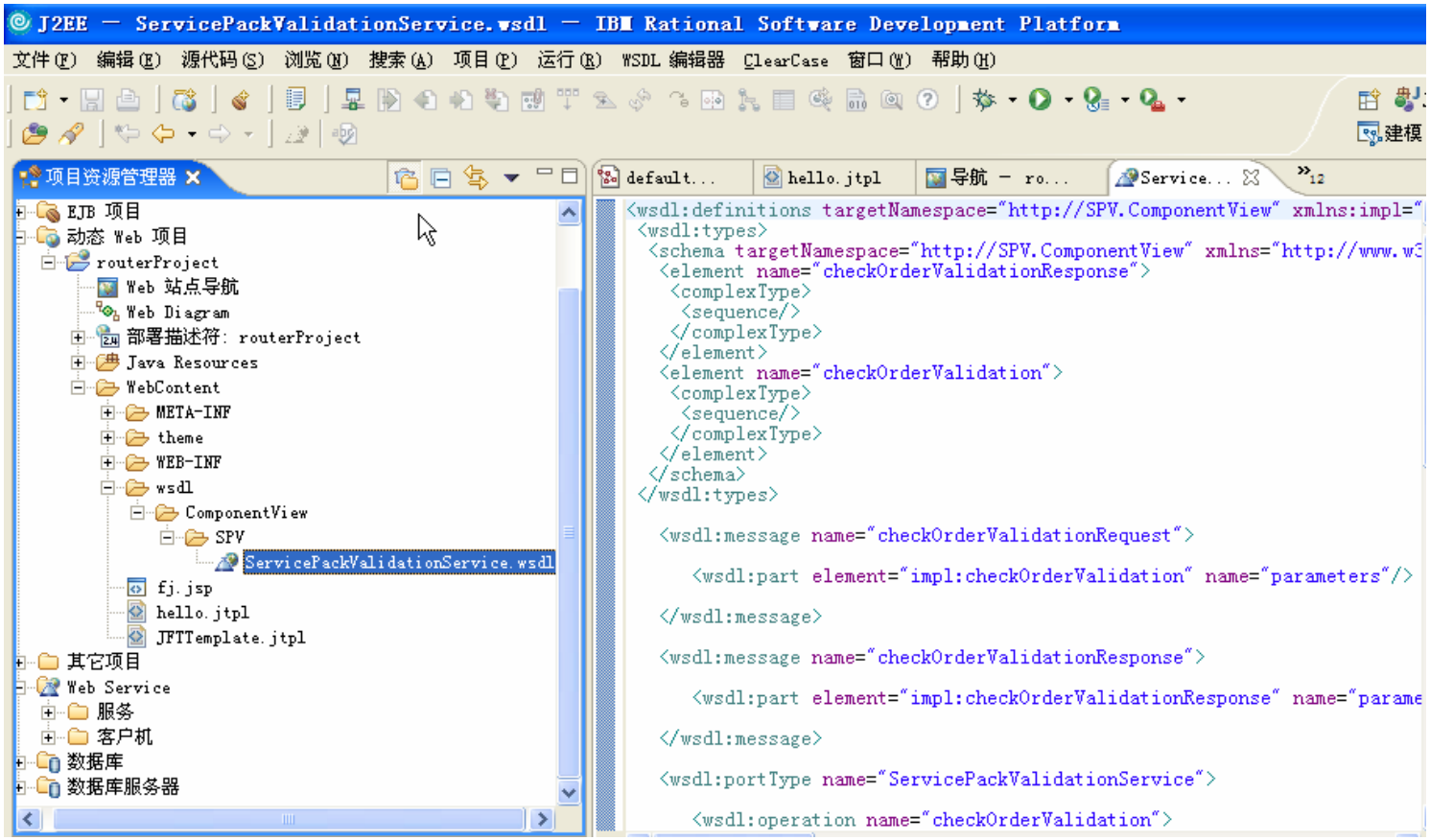
Demo8 - Bottom Up – 从EJB创建Web Service



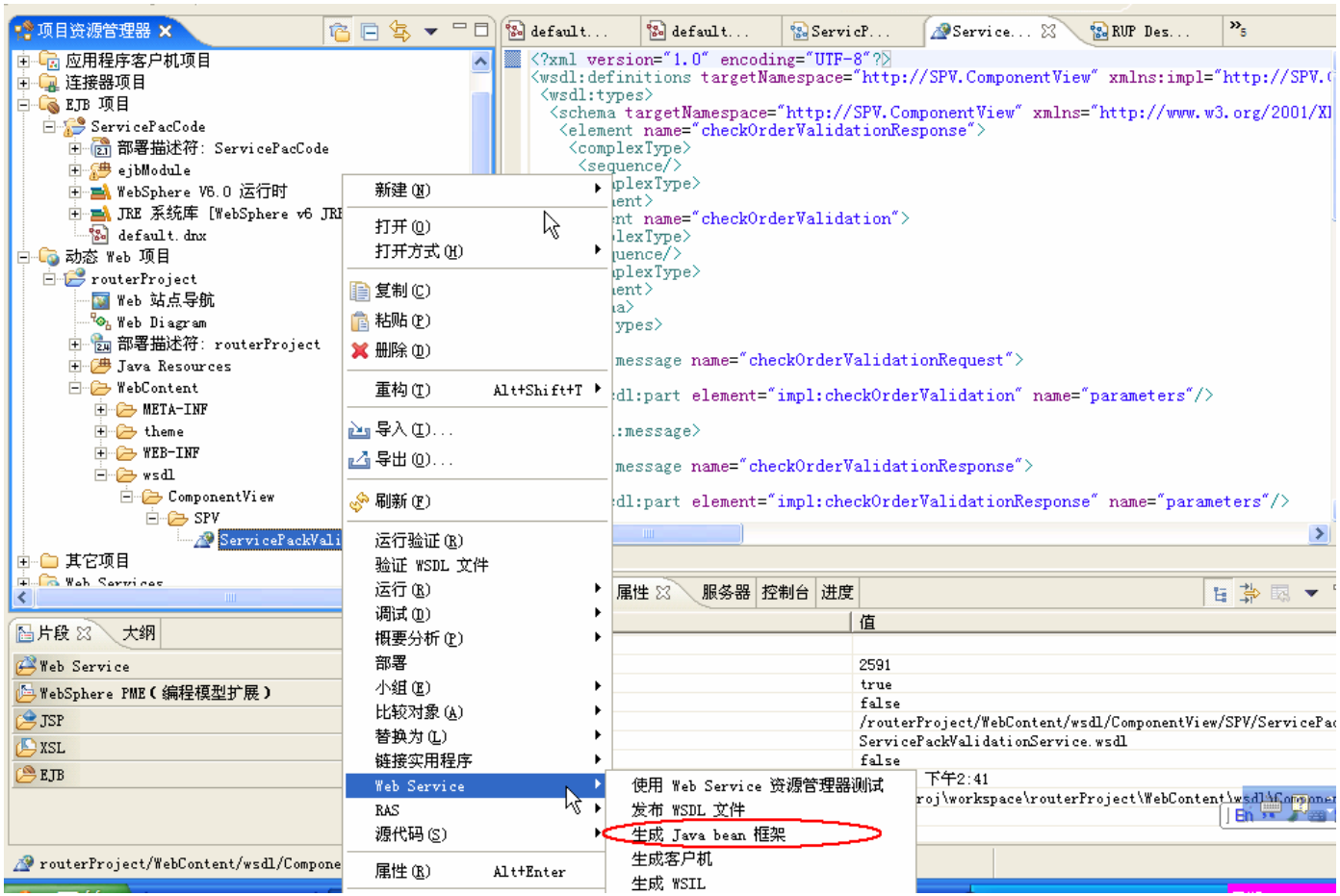
Bottom Up – 从EJB创建Web Service



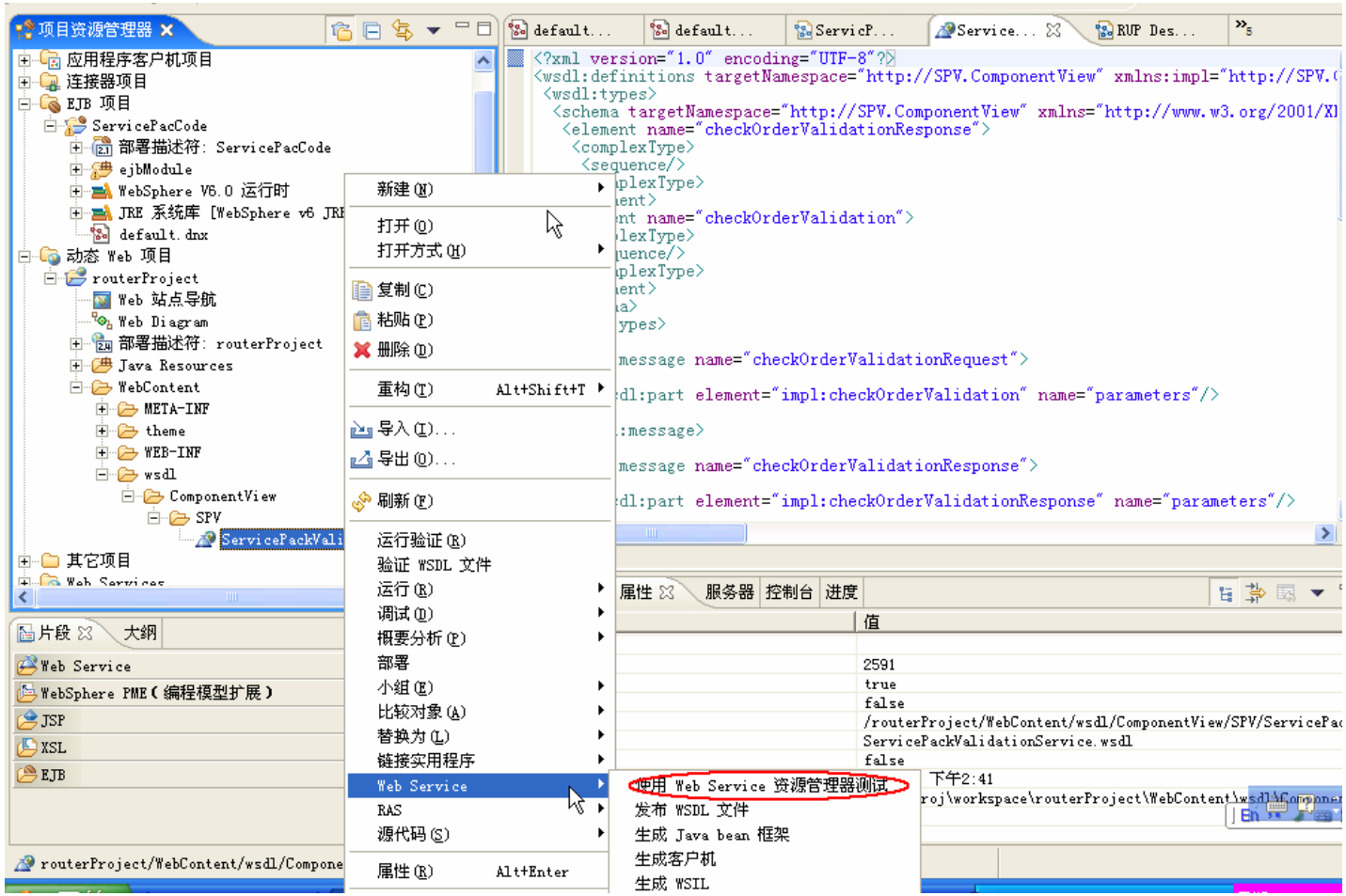
Bottom Up – 从EJB创建Web Service



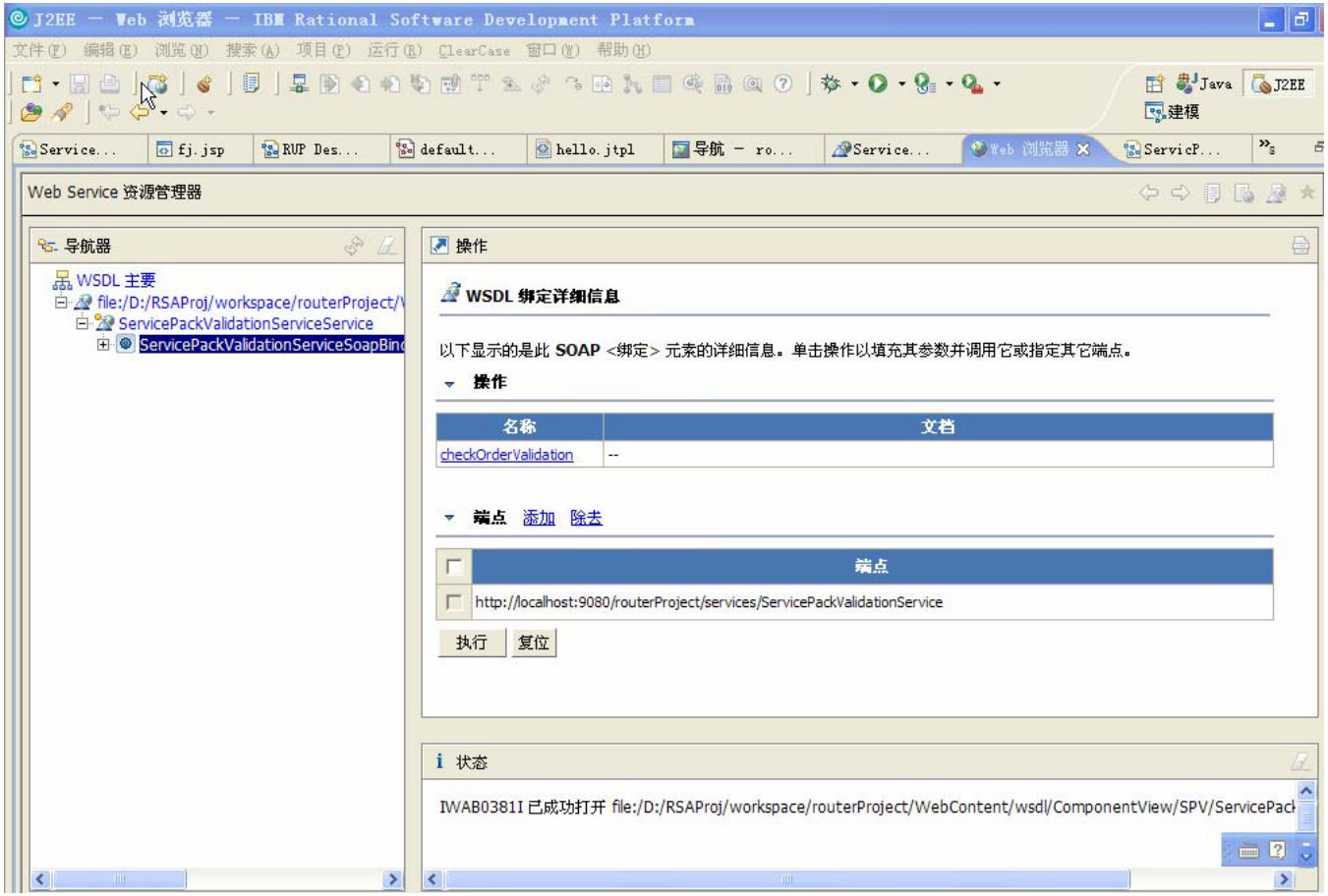
Top Down –从WSDL自动生成JAVA调用的代码



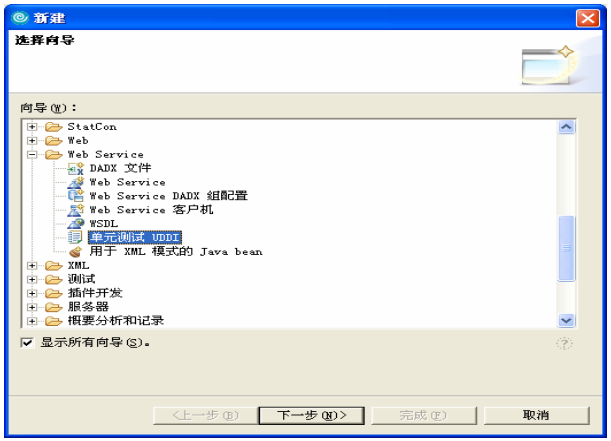
使用WSDL 浏览器进行Web Service执行测试

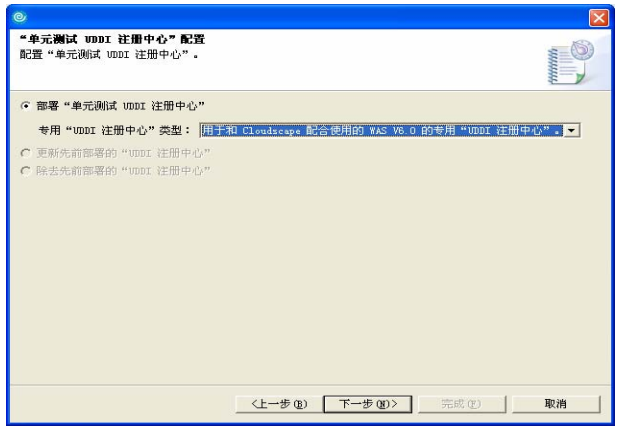


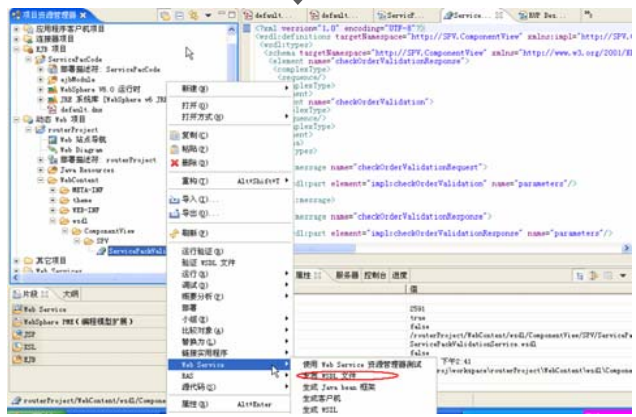
使用WSDL 浏览器进行Web Service执行测试

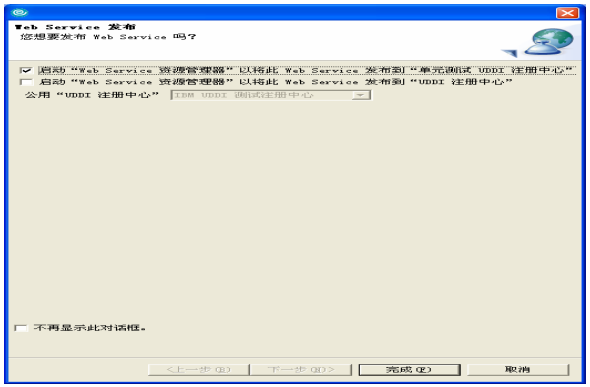


创建UDDI注册器并发布WSDL









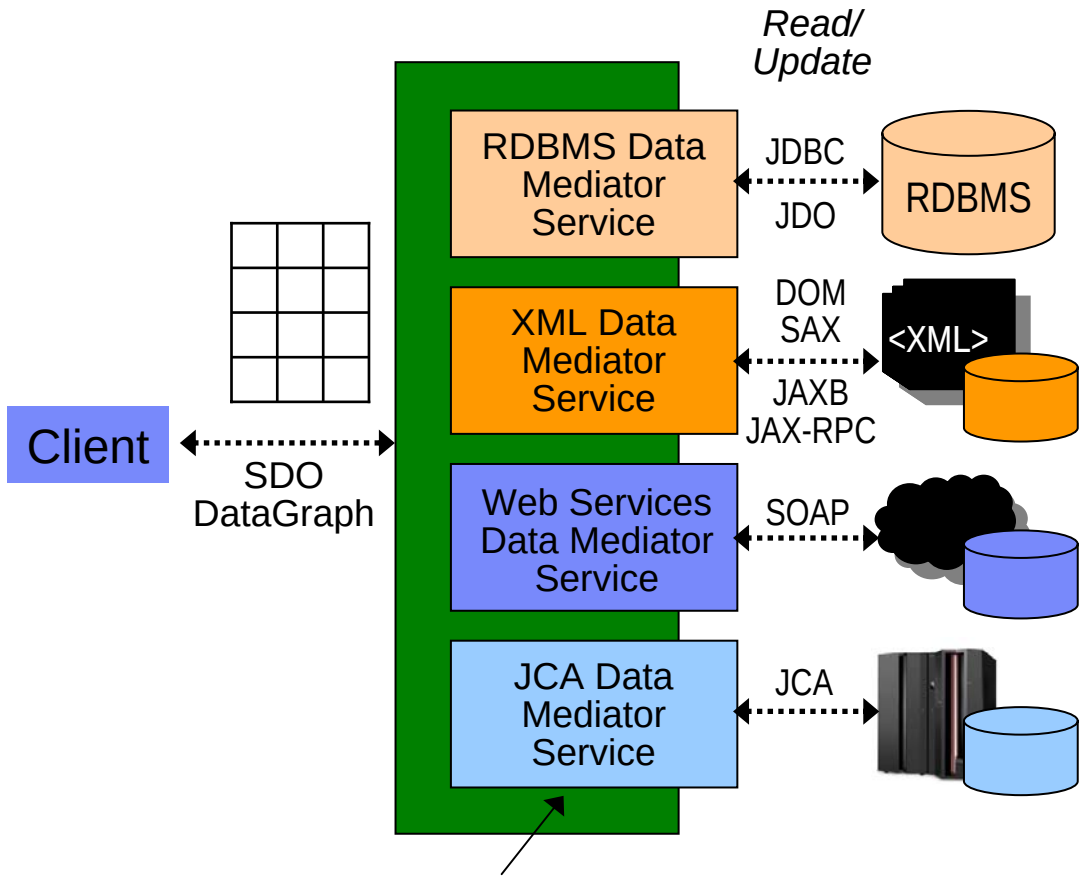
构建服务组件 - SOA开发第三步

- 快速构建Web Service
- 快速构建SDO
- 快速构建JSF

Service Data Objects (SDO): 统一的访问异构数据源

Emerging Standard

- 提供了一种通用的编程接口，用来访问异构的数据源
- 同时支持静态和动态的数据API
- 支持离线操作
- 将逻辑代码和数据访问代码剥离
- 用于提高程序员的开发效率

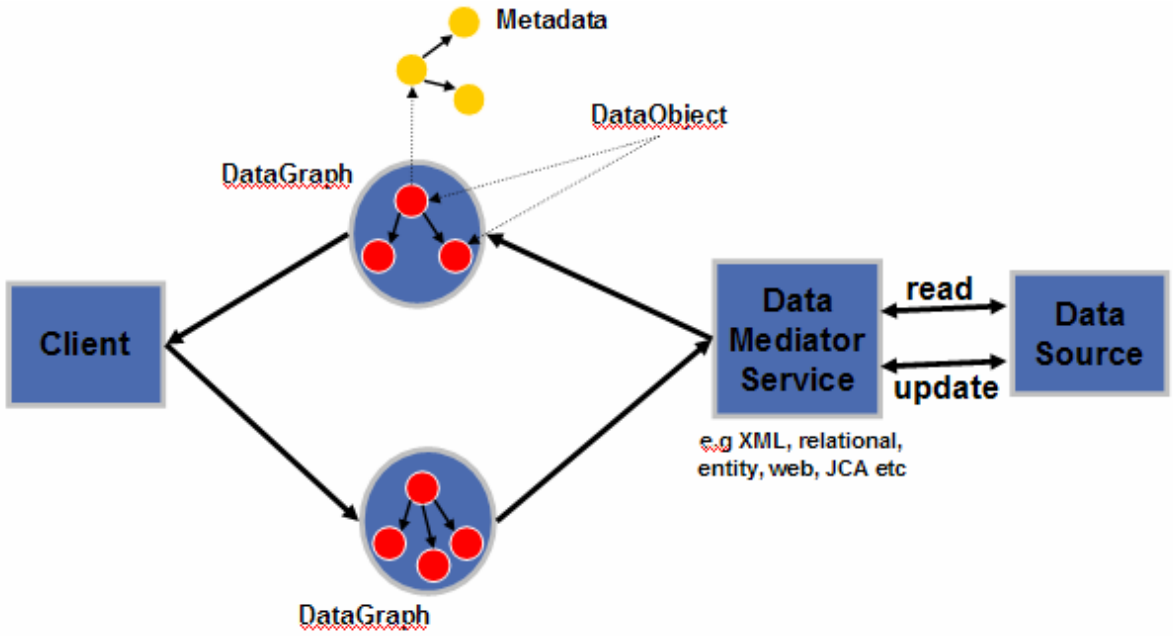


Service Data Objects (SDO) is a specification created by IBM & BEA, submitted to JCP as JSR 235 in December 2003. Expert Group Formation Stage, not part of J2EE yet

SDO的好处

SDO带来的好处：

- 统一数据应用开发
- 简化J2EE应用层对数据层的访问
- 支持和集成各种数据源
- J2EE pattern和最佳实践的体现



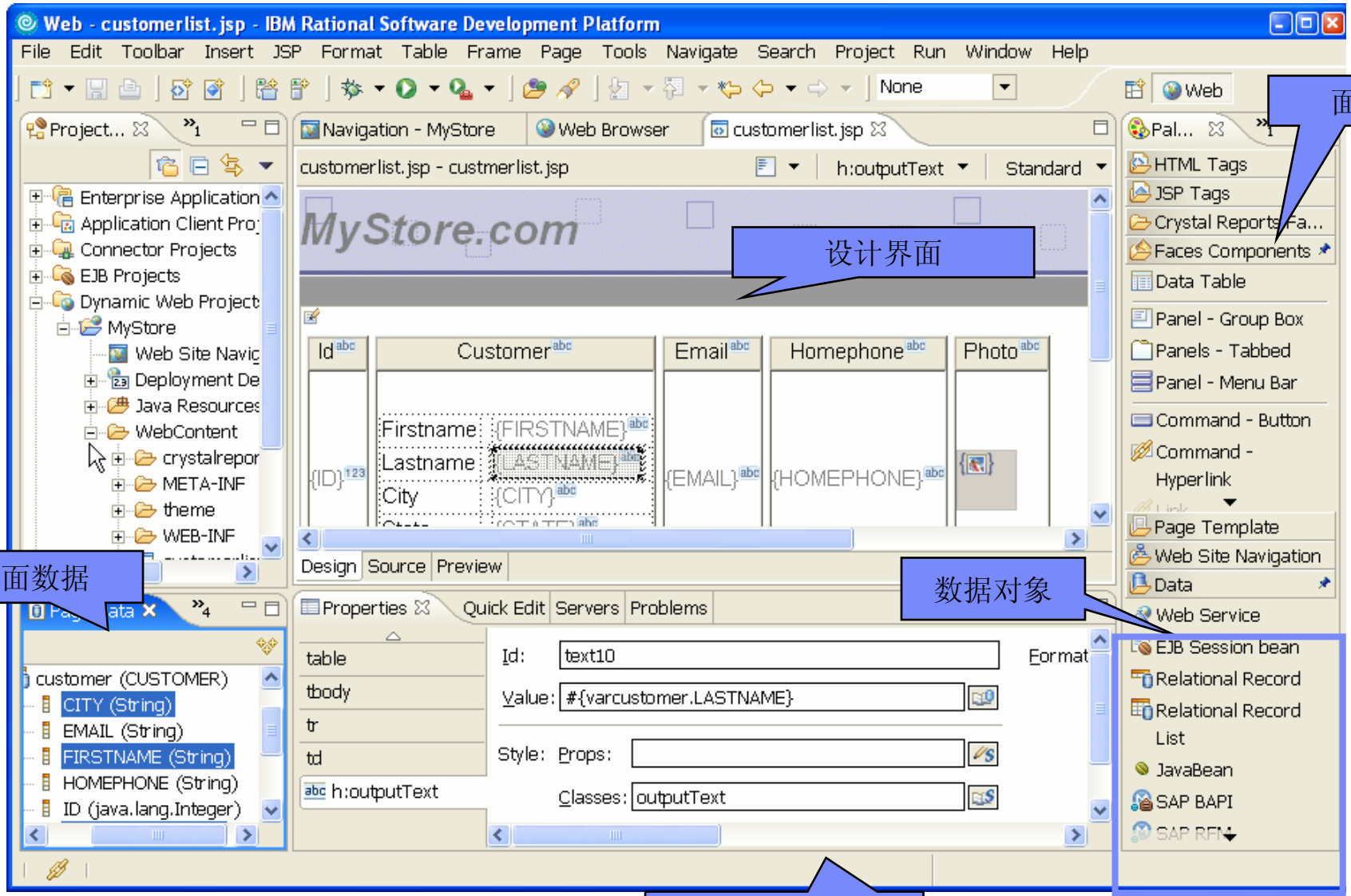
在RAD中构建SDO的两种方式

- 通过JSF
- 通过Entity Bean

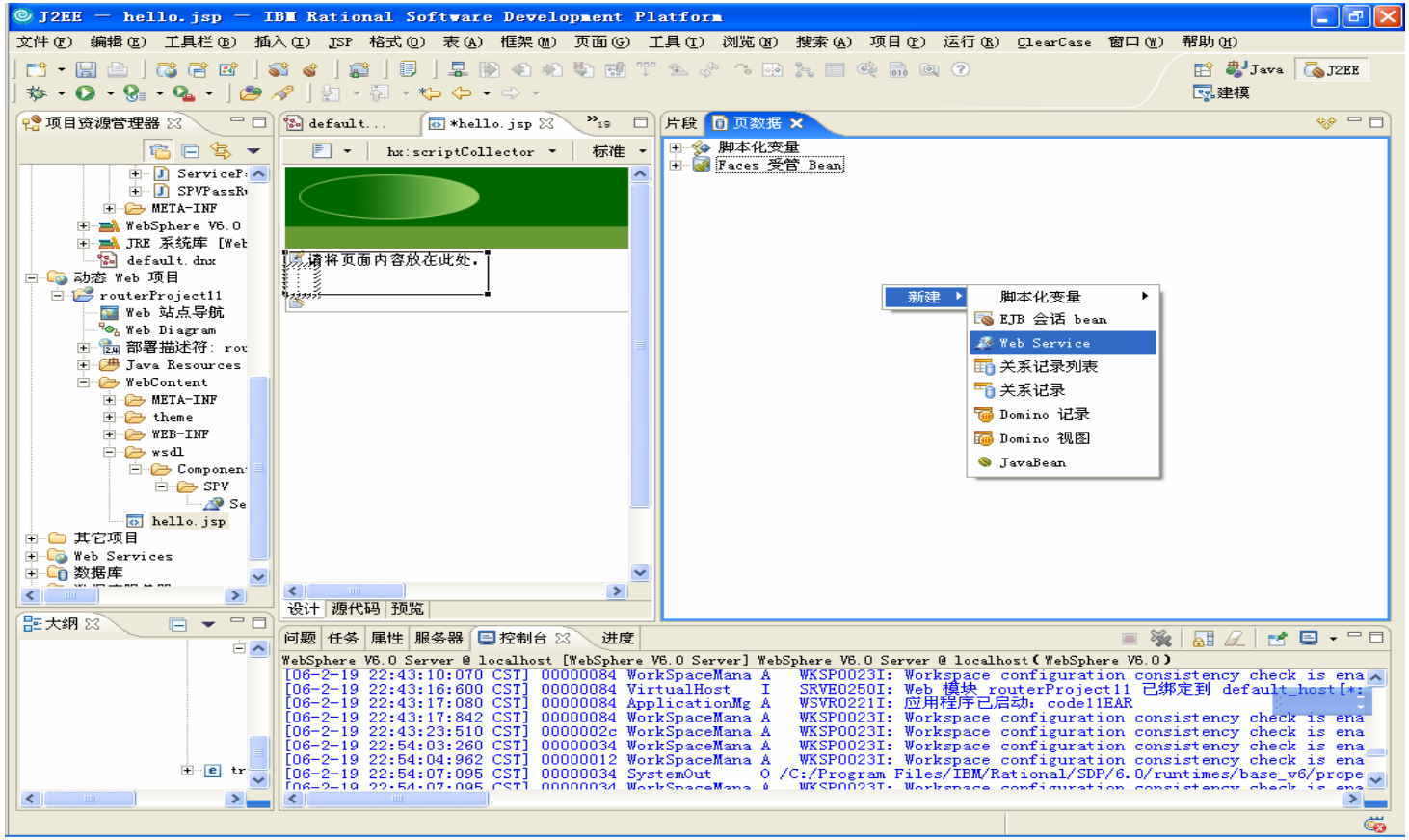
在RAD中构建SDO的两种方式

- 通过JSF
- 通过Entity Bean

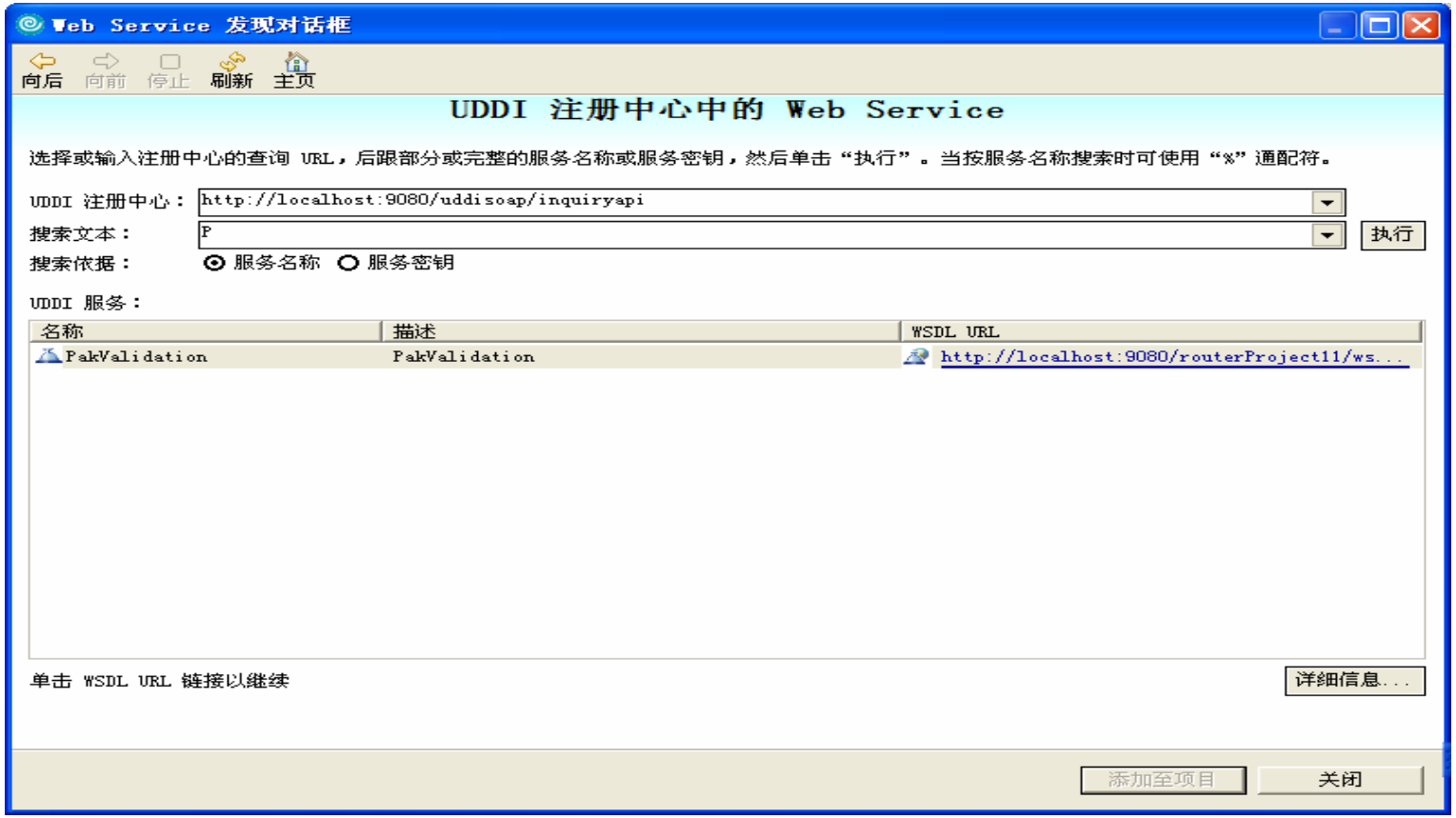
进行JSF/SDO的开发



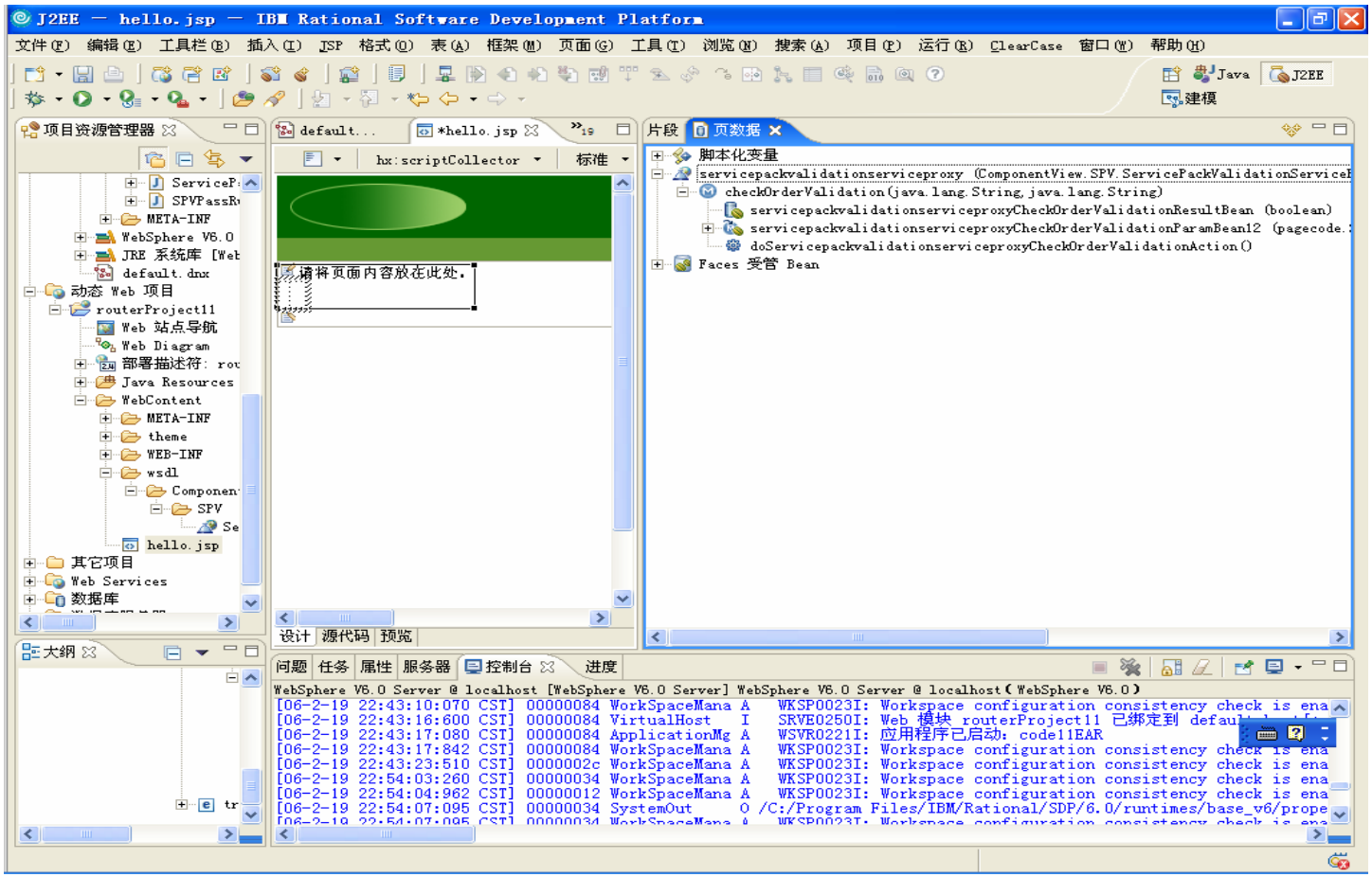
Demo9-使用SDO连接Web Service



使用SDO连接Web Service



使用SDO连接Web Service

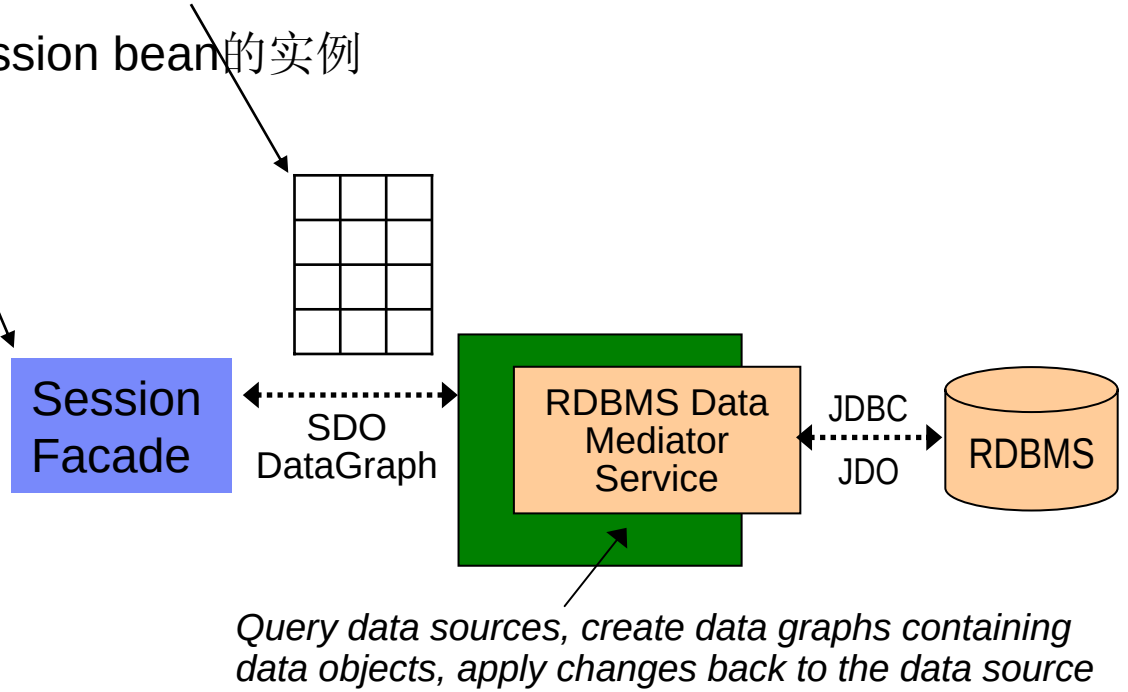


在RAD中构建SDO的两种方式

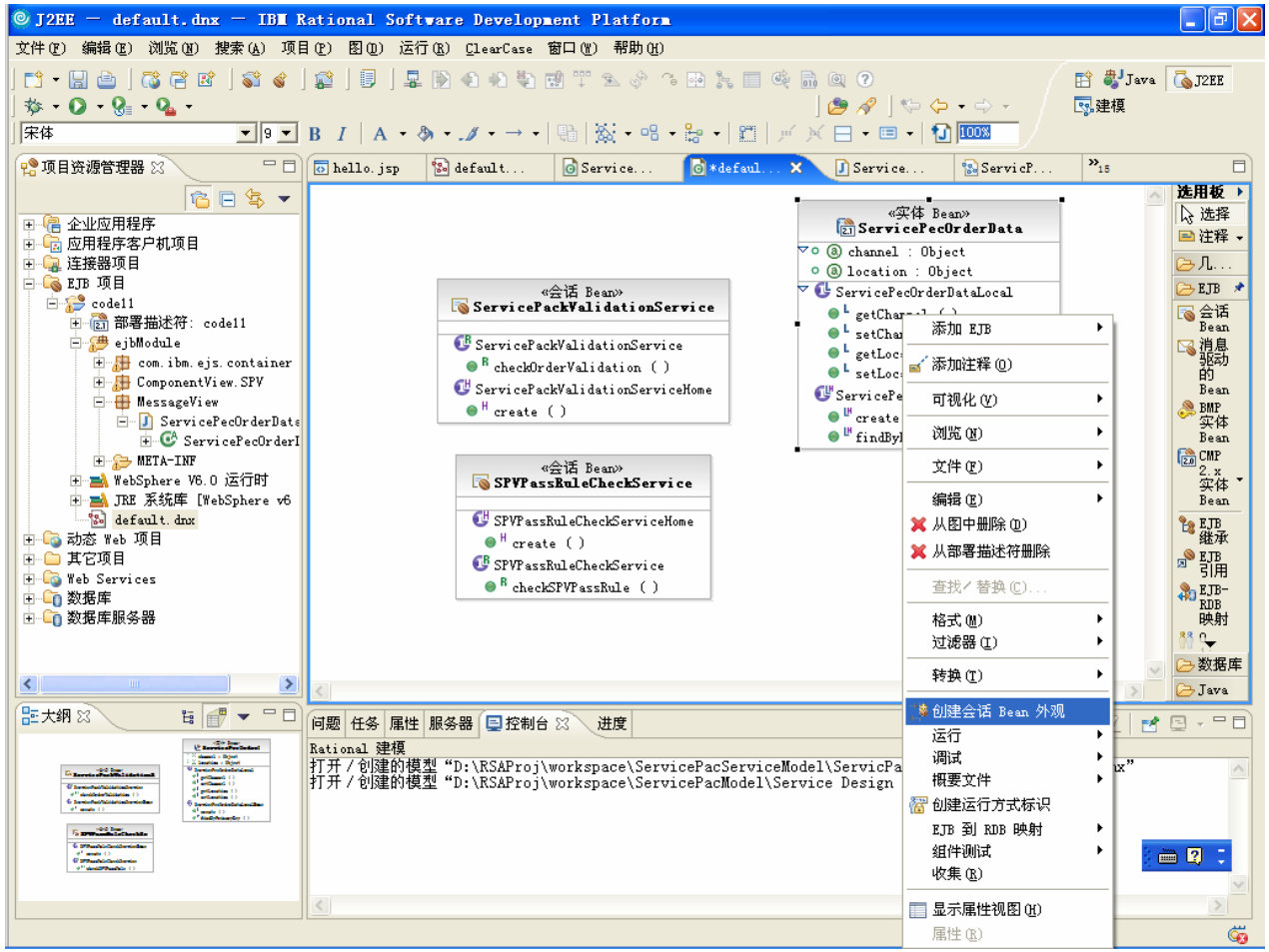
- 通过JSF
- 通过Entity Bean

EJB session façade及SDO 的创建

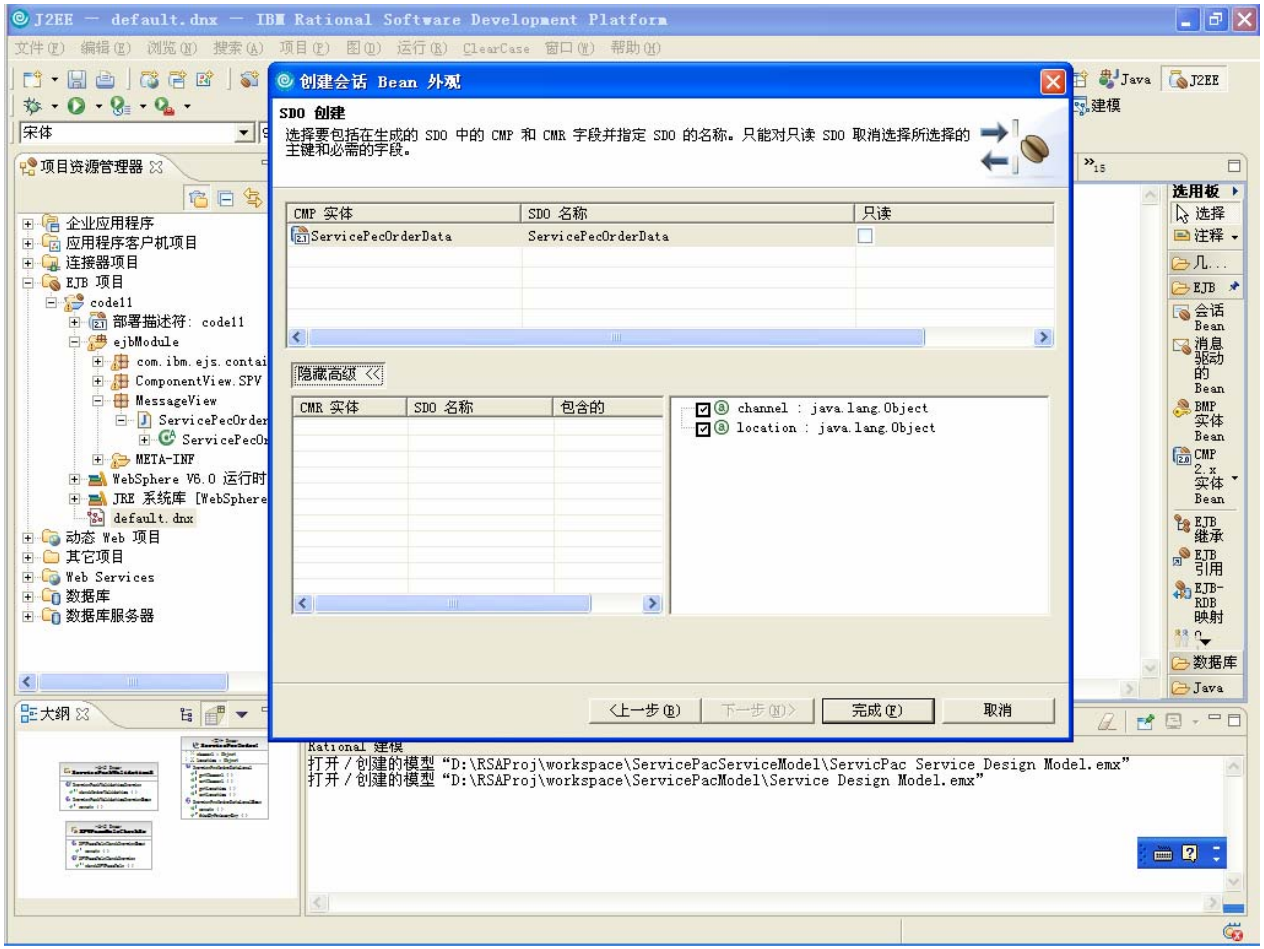
- 创建session façade bean
  - 通过CMP映射
  - 封装在一个或多个CMP实体bean中的数据和逻辑属性
  - 同时创建相应的SDO
- 结果是引用SDO的Session bean的实例



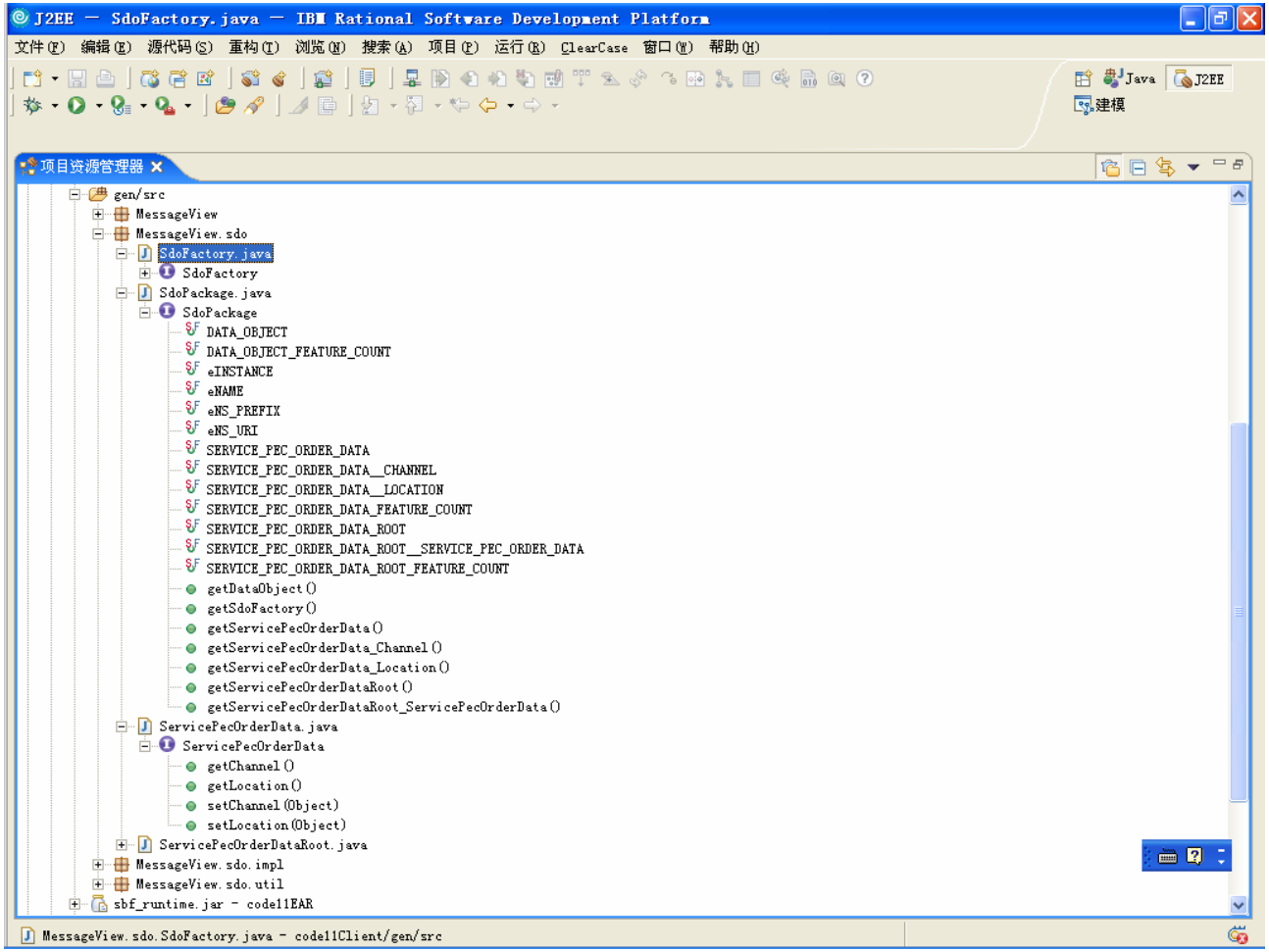
Demo10-根据CMP产生SDO



根据CMP产生SDO


J2EE — default.dnx — IBM Rational Software Development Platform
创建会话 Bean 外观
SDO 创建
选择要包括在生成的 SDO 中的 CMP 和 CMR 字段并指定 SDO 的名称。只能对只读 SDO 取消选择所选择的主键和必需的字段。
CMP 实体
SDO 名称
只读
ServicePecOrderData
ServicePecOrderData
隐藏高级 <<
CMR 实体
SDO 名称
包含的
channel : java.lang.Object
location : java.lang.Object
<上一步(B)
下一步(N)>
完成(F)
取消
项目资源管理器
企业应用程序
应用程序客户机项目
连接器项目
EJB 项目
code11
部署描述符: code11
ejbModule
META-INF
动态 Web 项目
其它项目
Web Services
数据库
数据库服务器
大纲
Rational 建模
打开 / 创建的模型 "D:\RSAProj\workspace\ServicePacServiceModel\ServicPac Service Design Model.emx"
打开 / 创建的模型 "D:\RSAProj\workspace\ServicePacModel\Service Design Model.emx"

根据CMP产生SDO


J2EE — SdoFactory.java — IBM Rational Software Development Platform
文件(F) 编辑(E) 源代码(S) 重构(T) 浏览(N) 搜索(A) 项目(P) 运行(R) ClearCase 窗口(W) 帮助(H)
Java
J2EE
建模
项目资源管理器
gen/src
MessageView
MessageView.sdo
SdoFactory.java
SdoFactory
SdoPackage.java
SdoPackage
DATA_OBJECT
DATA_OBJECT_FEATURE_COUNT
eINSTANCE
eNAME
eNS_PREFIX
eNS_URI
SERVICE_PEC_ORDER_DATA
SERVICE_PEC_ORDER_DATA__CHANNEL
SERVICE_PEC_ORDER_DATA__LOCATION
SERVICE_PEC_ORDER_DATA_FEATURE_COUNT
SERVICE_PEC_ORDER_DATA_ROOT
SERVICE_PEC_ORDER_DATA_ROOT__SERVICE_PEC_ORDER_DATA
SERVICE_PEC_ORDER_DATA_ROOT_FEATURE_COUNT
getDataObject()
getSdoFactory()
getServicePecOrderData()
getServicePecOrderData_Channel()
getServicePecOrderData_Location()
getServicePecOrderDataRoot()
getServicePecOrderDataRoot_ServicePecOrderData()
ServicePecOrderData.java
ServicePecOrderData
getChannel()
getLocation()
setChannel(Object)
setLocation(Object)
ServicePecOrderDataRoot.java
MessageView.sdo.impl
MessageView.sdo.util
sbf_runtime.jar - code11EAR
MessageView.sdo.SdoFactory.java - code11Client/gen/src

构建服务组件 - SOA开发第三步

- 快速构建Web Service
- 快速构建SDO
- 快速构建JSF

JSF简介

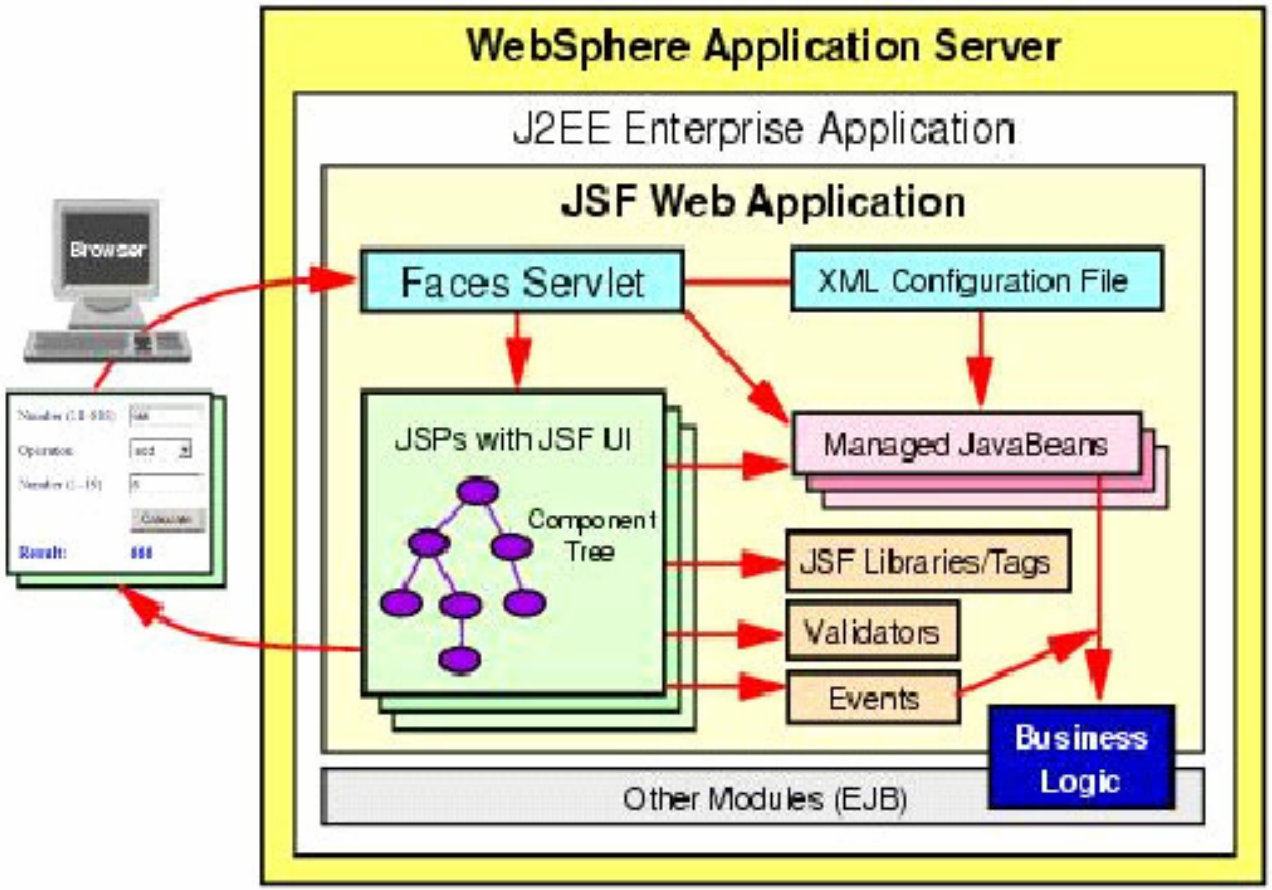
JSF为开发 Web应用用户界面提供了：

- 标准的编程接口
- 丰富可扩展的UI组件库
- 事件驱动模型等一套完整的Web应用框架

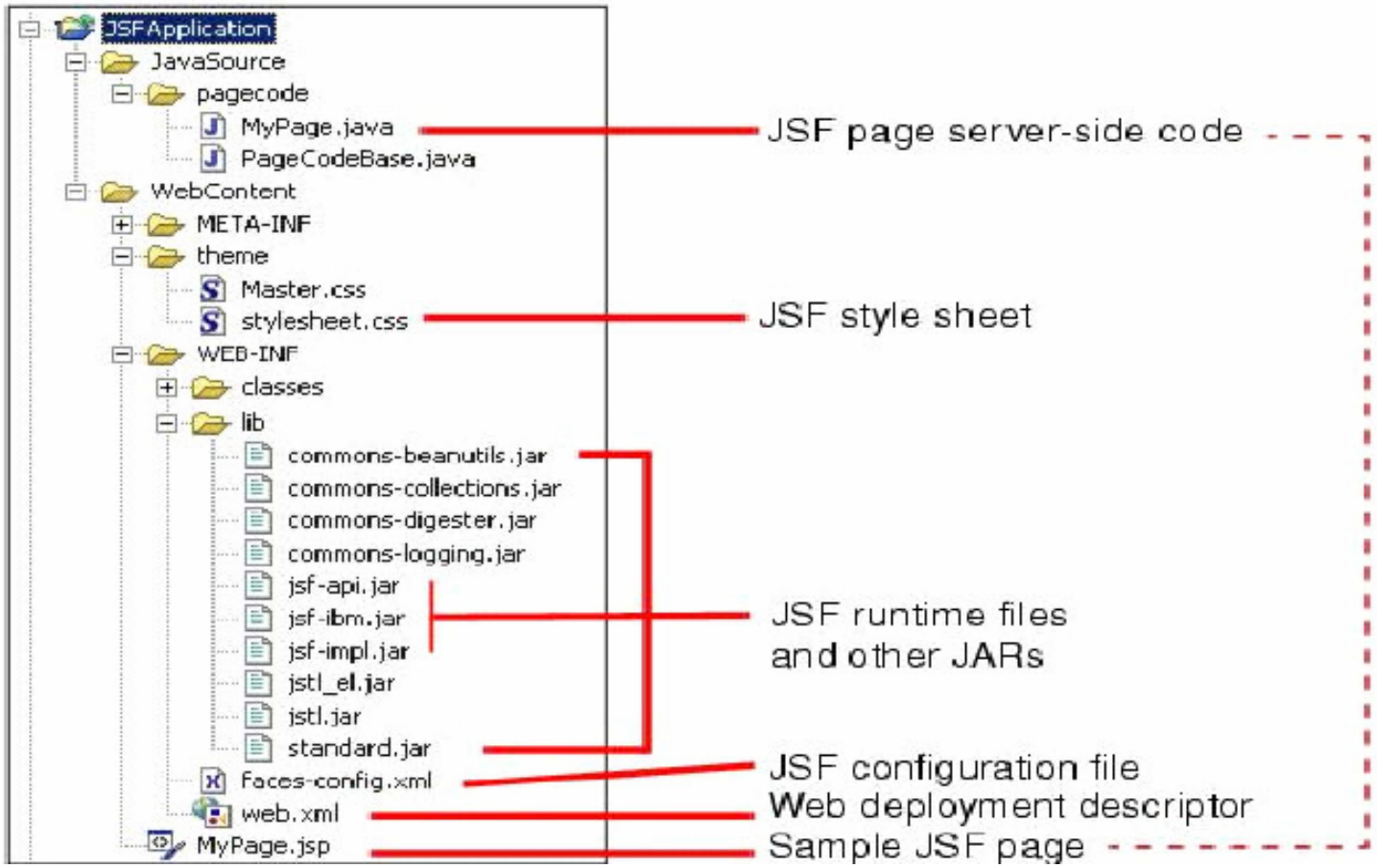
通过 JSF ， 开发人员在页面中轻松自如地使用 WEB 组件、捕获用户行为所产生的事件、执行验证、建立页面导航等

JSF:

JSF框架

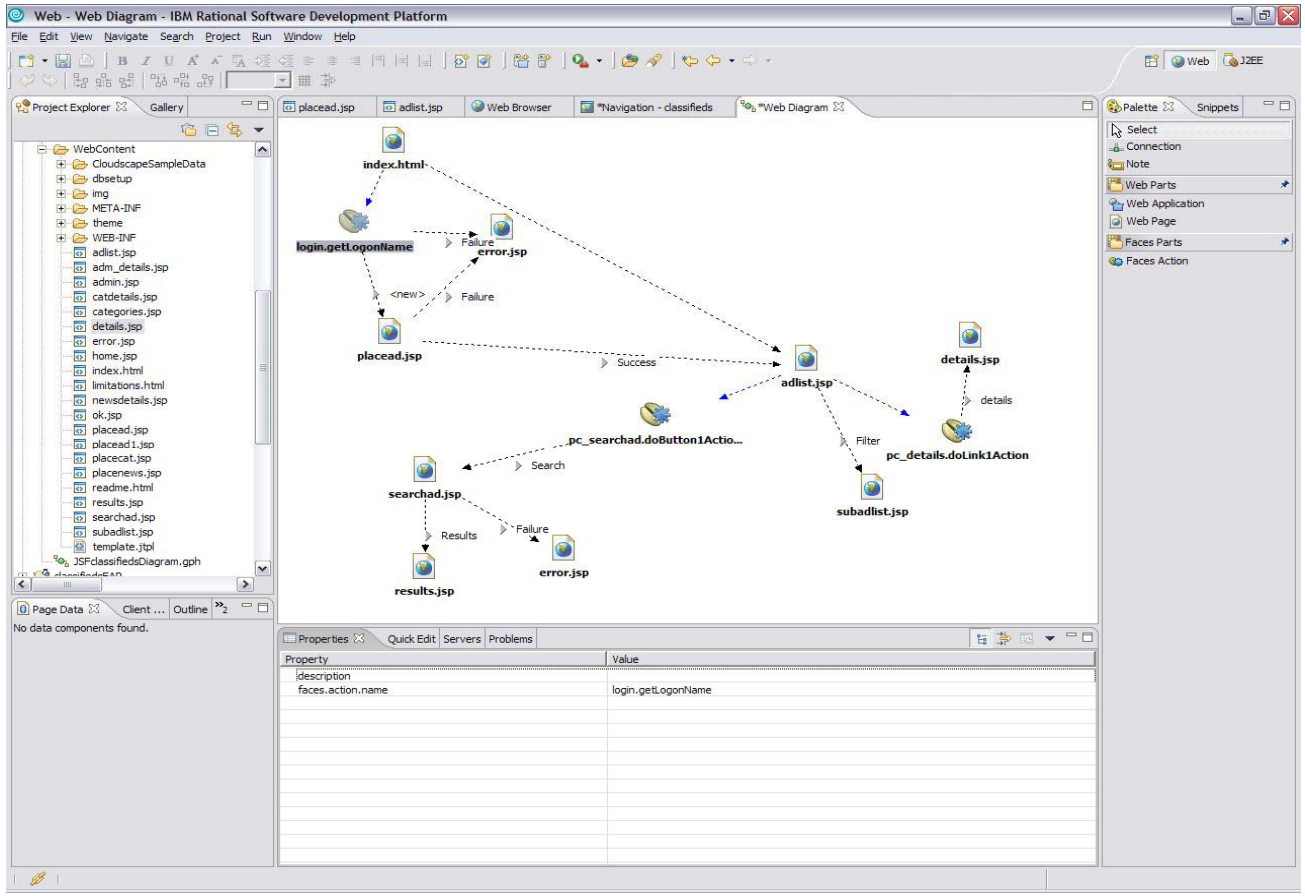

WebSphere Application Server
J2EE Enterprise Application
JSF Web Application
Browser
Faces Servlet
XML Configuration File
JSPs with JSF UI
Component Tree
Managed JavaBeans
JSF Libraries/Tags
Validators
Events
Business Logic
Other Modules (EJB)

RAD开发JSF－目录结构



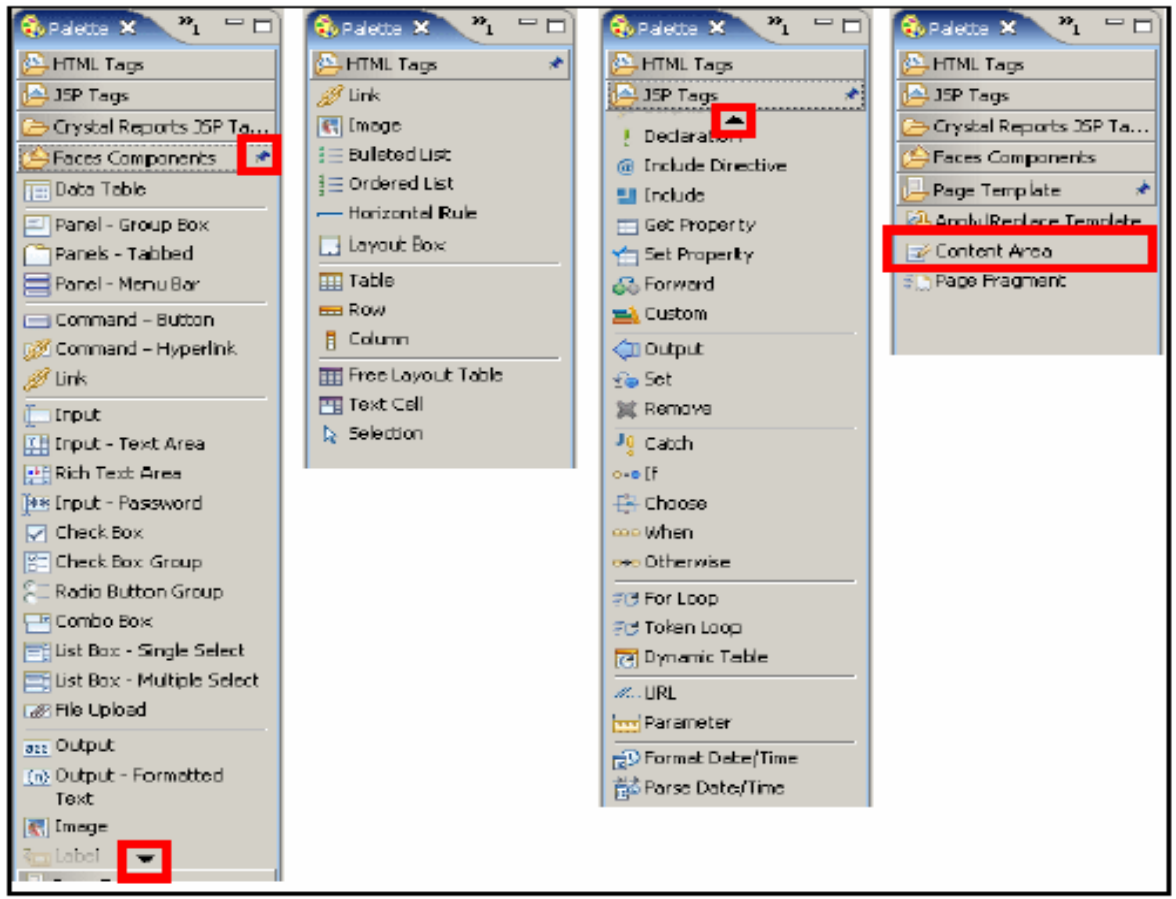
RAD开发JSF－Web Diagram

通过Web Diagram tool 创建JSF资源如JSF pages, JSPs, Faces actions, 和pages与actions之间的连接



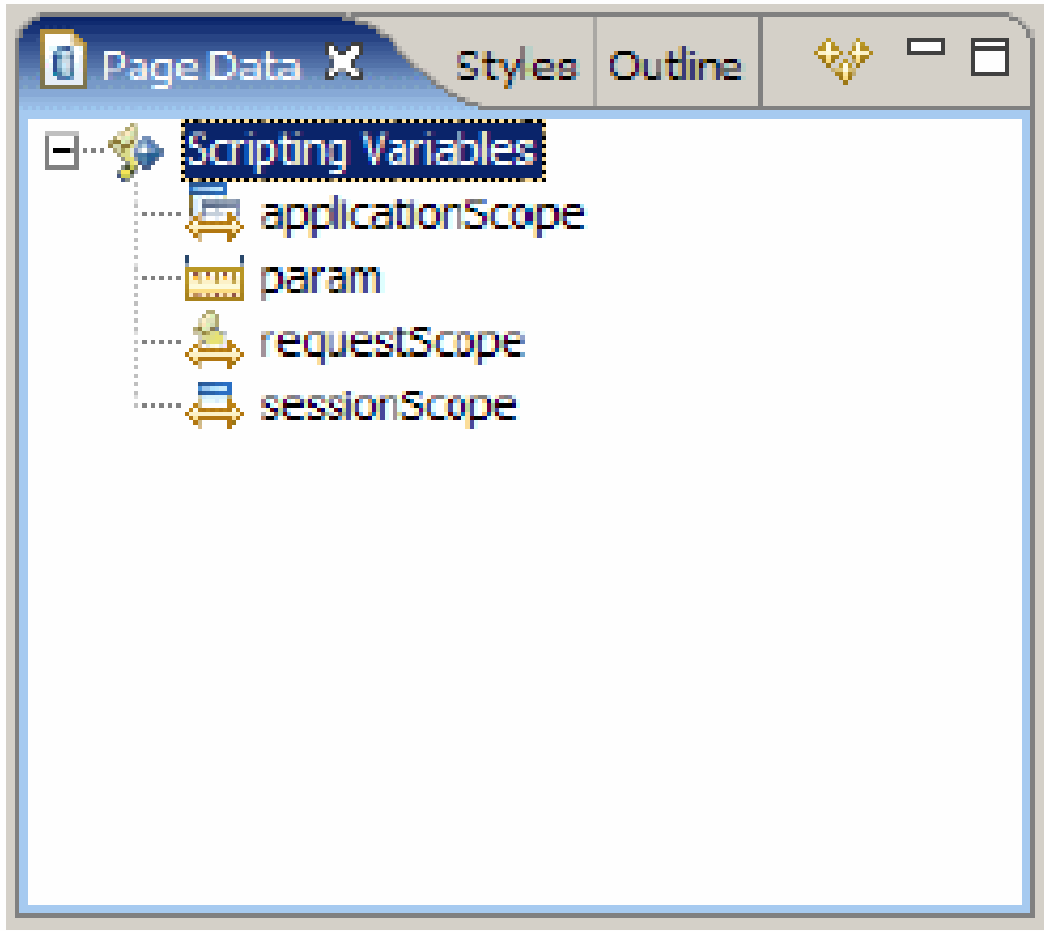
RAD开发JSF－Palette

Palette包括一些常用的组件如JSP tags，HTML tags和Face components等等，可以直接拖拽到JSF页面上



RAD开发JSF－Page Data

可以将各种数据源如Web Service, Database(SDO)和EJB Session Bean等加入到页面中




Thank
You