IBM Software Group

SOA 开发第三步 – 构建服务组件

IBM 软件部 李傲雷

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

- 精练服务架构
- 设计服务
- 设计组件

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

组件框架
组件代码

- 生成组件框架
- 编码实现组件

Rational. software
Rational Application Developer

资源组合　共享资产　公共流程

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

内容

- 快速构建Web Service
- 快速构建JSF＋SDO

内容

- 快速构建Web Service
- 快速构建JSF＋SDO

快速构建Web Service－ Web Service简介

- 什么是**Web Service**？
  - Web Service是分布式处理技术，是对象/组件技术在Internet中的延伸，是封装成单个实体且发布到网络上以供其它程序使用的功能集合
  - Web Service从本质上讲是放置于Web上的可重用构件
  - Web Service可以分散于Web的各个地方，通过互相地调用以协同完成业务活动
  - 在Web Service的体系中，应用系统被分割为高内聚、松耦合的单个的服务，可以通过Web被调用和访问
  - Web Service核心基础是扩展标记语言XML，其相关标准协议包括服务调用协议SOAP、服务描述语言WSDL以及服务注册检索访问标准UDDI

快速构建Web Service— Web Service简介

- SOAP - Simple Object Access Protocol
  - 以XML技术为基础，基于各种传输协议进行消息传递
  - SOAP主要定义了4个元素：SOAP封套、数据编码规则、绑定约定、RPC约定
- WSDL - Web Services Description Language
  - 用于解决客户端如何才能知道Web Service提供了怎样的可调用服务
  - 物理上是一个XML文件，用于说明SOAP消息以及如何交换这些消息
- UDDI - Universal Description, Discovery, Integration
  - UDDI 是一个跨产业、跨平台的开放体系结构，其核心是UDDI商业注册
  - UDDI商业注册使用一个XML文档来描述企业及其提供的Web Service
  - 借助UDDI，Web Service开发商才能把自己开发的Web Service发布到Inrternet上供客户选择和使用
- 用一句话概括：
  - 用WDSL对服务进行描述和说明，通过UDDI对服务进行注册，通过SOAP在Web上访问使用服务

快速构建Web Service－ Web Service简介

- JAX-RPC - Java API for XML-Based RPC
  - 用于实现Web Service的互操作性，把 XML 类型转换为 Java 类型，以确保 Web Service客户端和基于 Java 的应用程序之间能够进行平稳的数据交换
  - 通过类型映射系统将WSDL文档中定义的各种XML类型映射为Java服务接口所指定的相应Java 类型，反之亦然，这是使 Web 服务可实现互操作的关键
  - 使服务器能够用标准的 API 定义其服务并且能够用 WSDL 描述其服务，使客户端能够用标准的 API 与服务器进行通信
- JSR-109

  JSR-109是J2EE的扩展规格文档之一，涵盖以下领域：
  - 客户端编程模式－J2EE如何把Web Service作为传统的远程对象来访问，即本地客户端如何访问Web Service
  - 服务器端编程模式－怎样用Servlets和无状态会话EJB来实现Web Service并使Web Service具有和他们一样的生命周期
  - Web Services部署和部署描述器－如何在应用服务器上部署Web Service

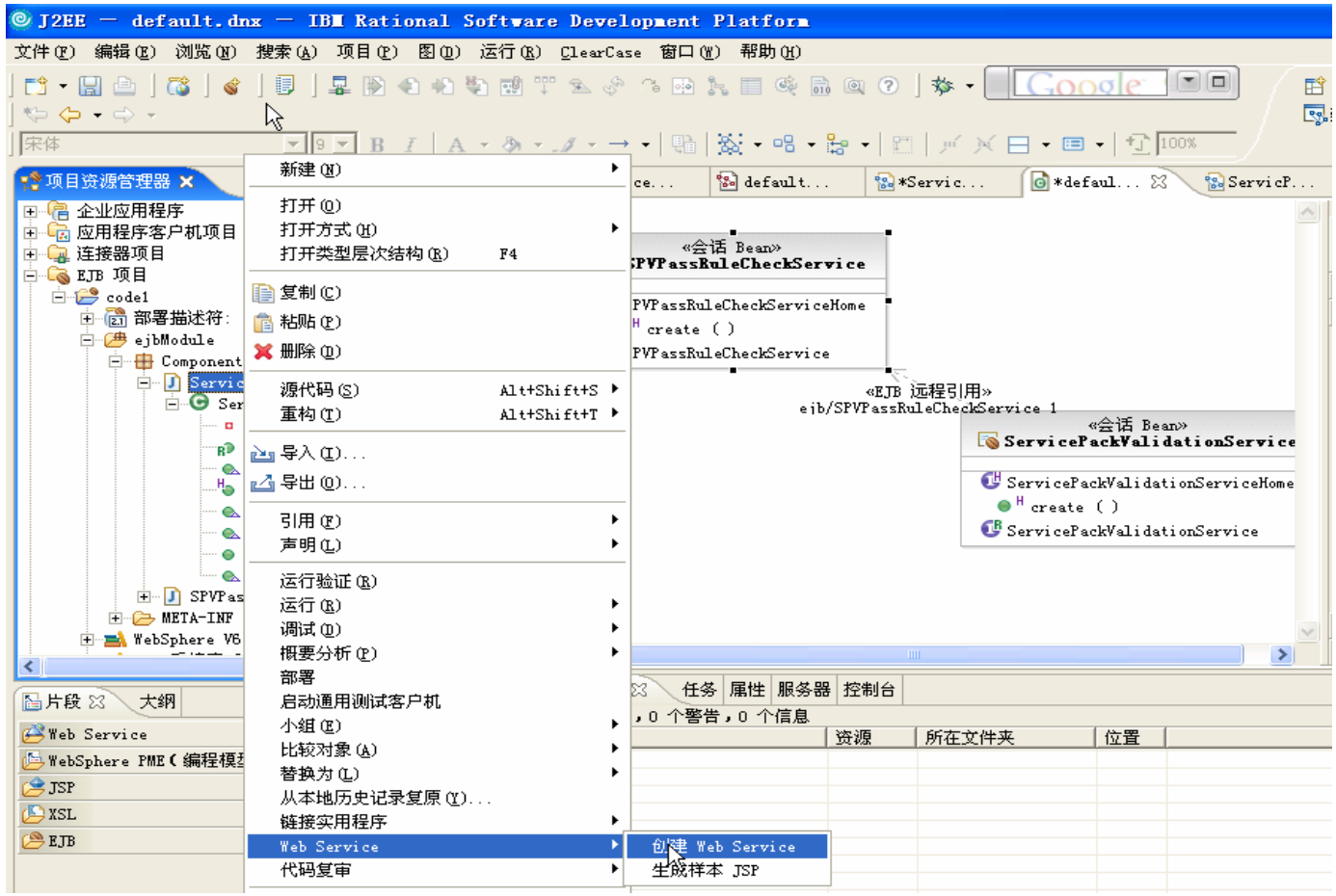
快速构建Web Service－ Web Service与SOA

- Web 服务则是基于标准的、可经济实惠地实现 SOA的一项技术
- SOA不是一定需要 Web 服务来实现，一个基于Web 服务开发出来的应用也不代表就是一个基于 SOA 构架应用
- Web 服务只是服务实现的一个典型，是实现企业 SOA的一个组件（非必需组件）
- SOA 为基于服务的分布式系统提供了概念上的设计模式

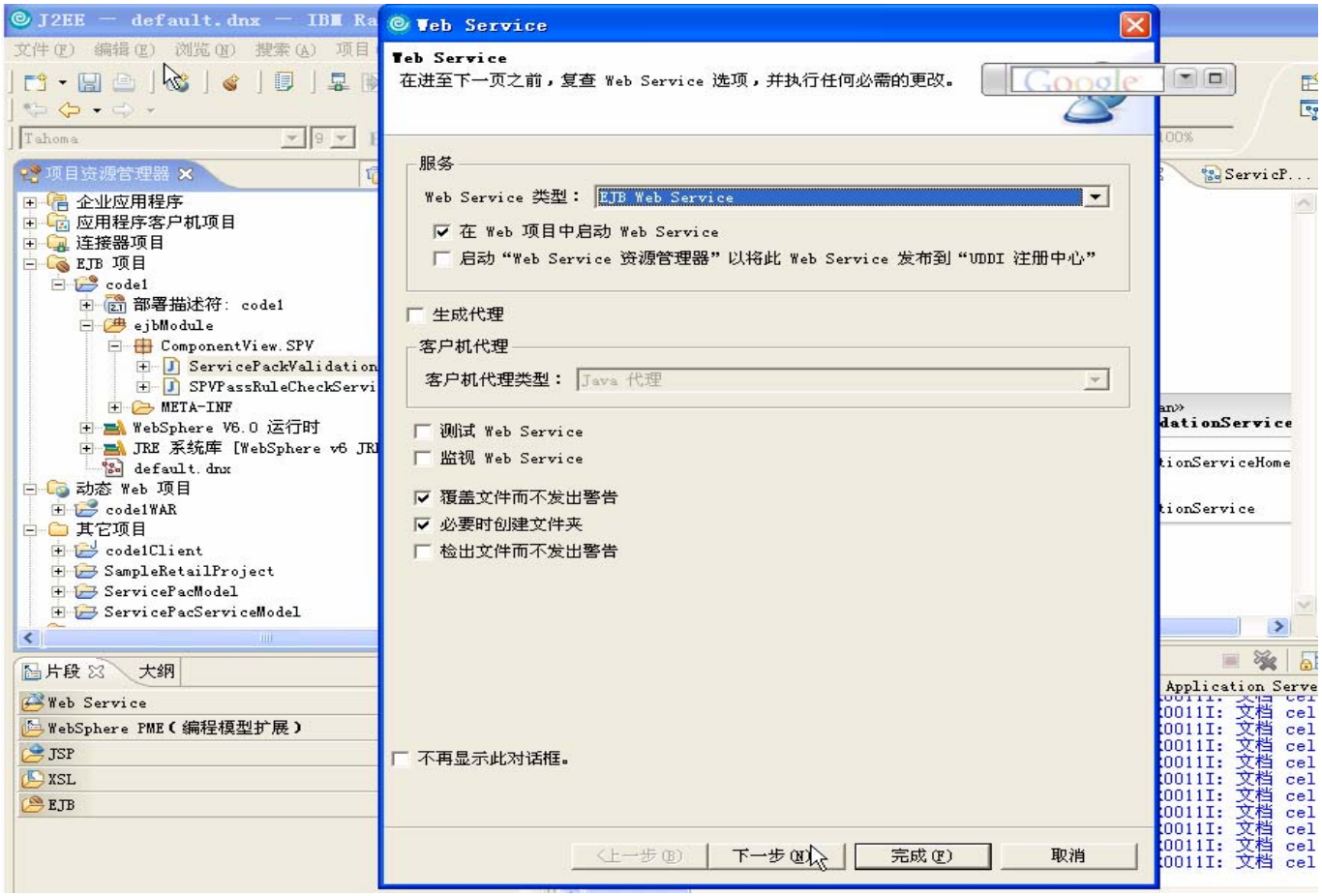
快速构建Web Service－RAD的开发能力

- 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 Soap环境
- 发布
  - 将Web services 发布到 UDDI v2 or v3 注册服务器上

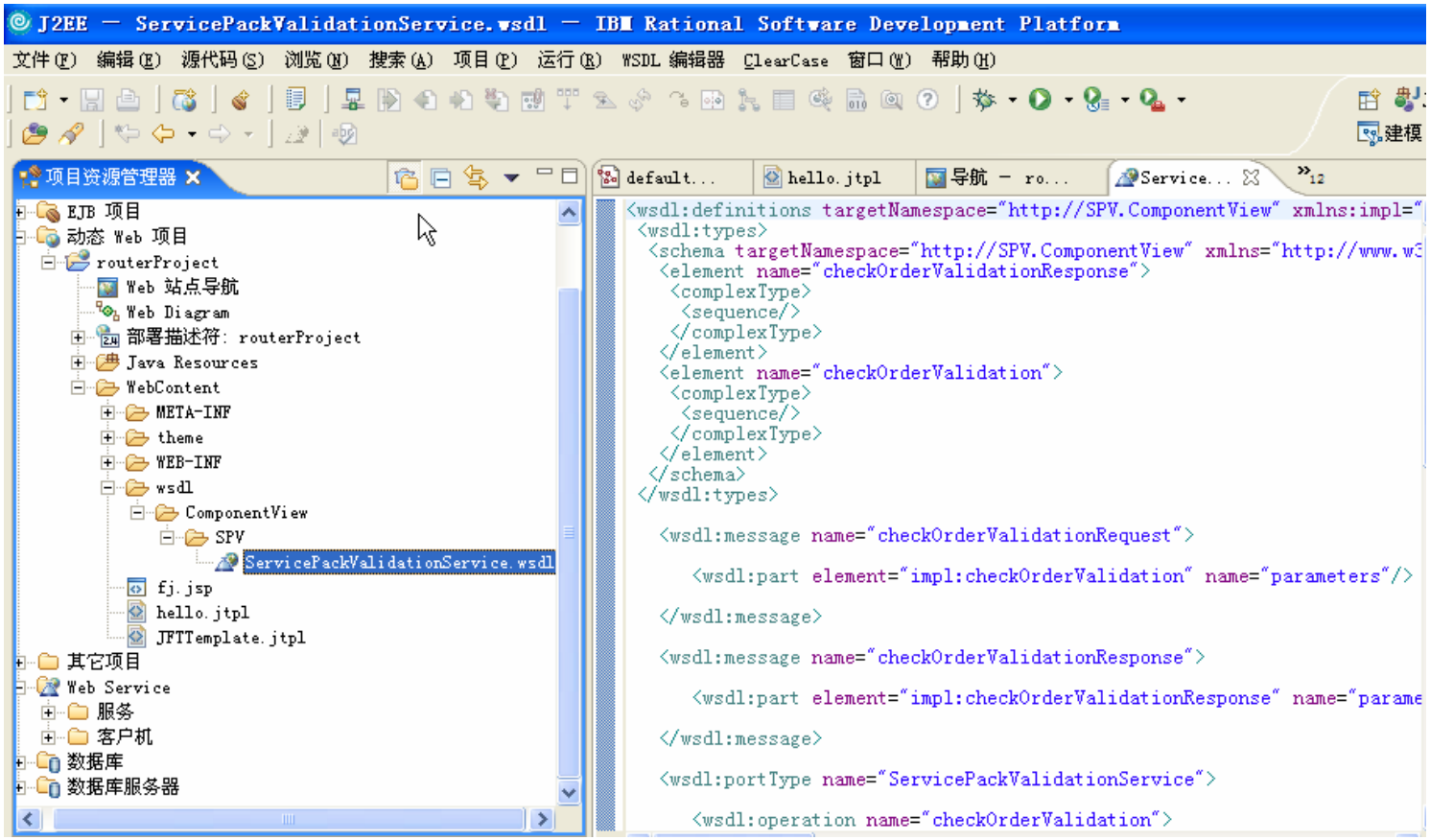
Demo8 - Bottom Up – 从EJB创建Web Service



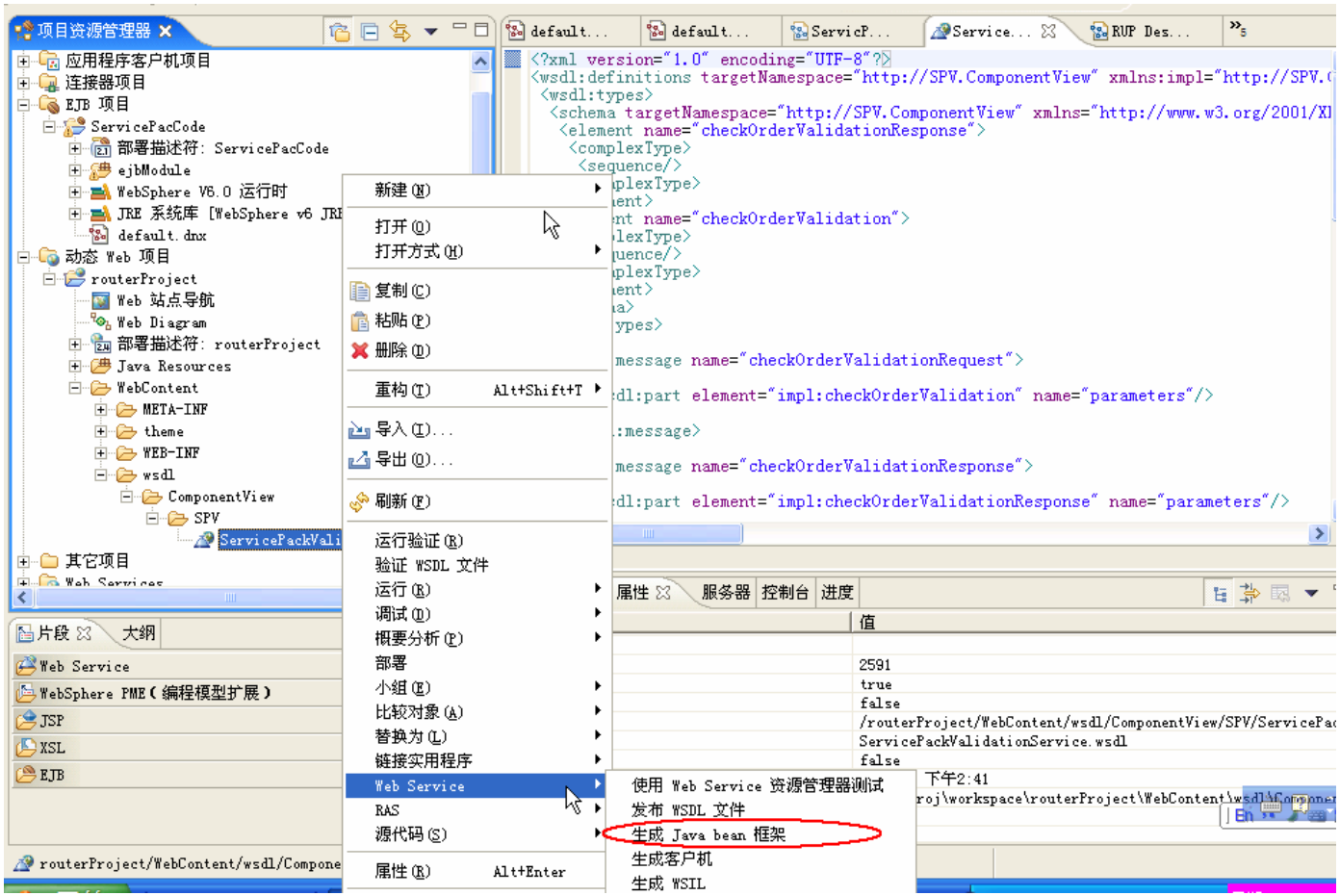
Bottom Up – 从EJB创建Web Service



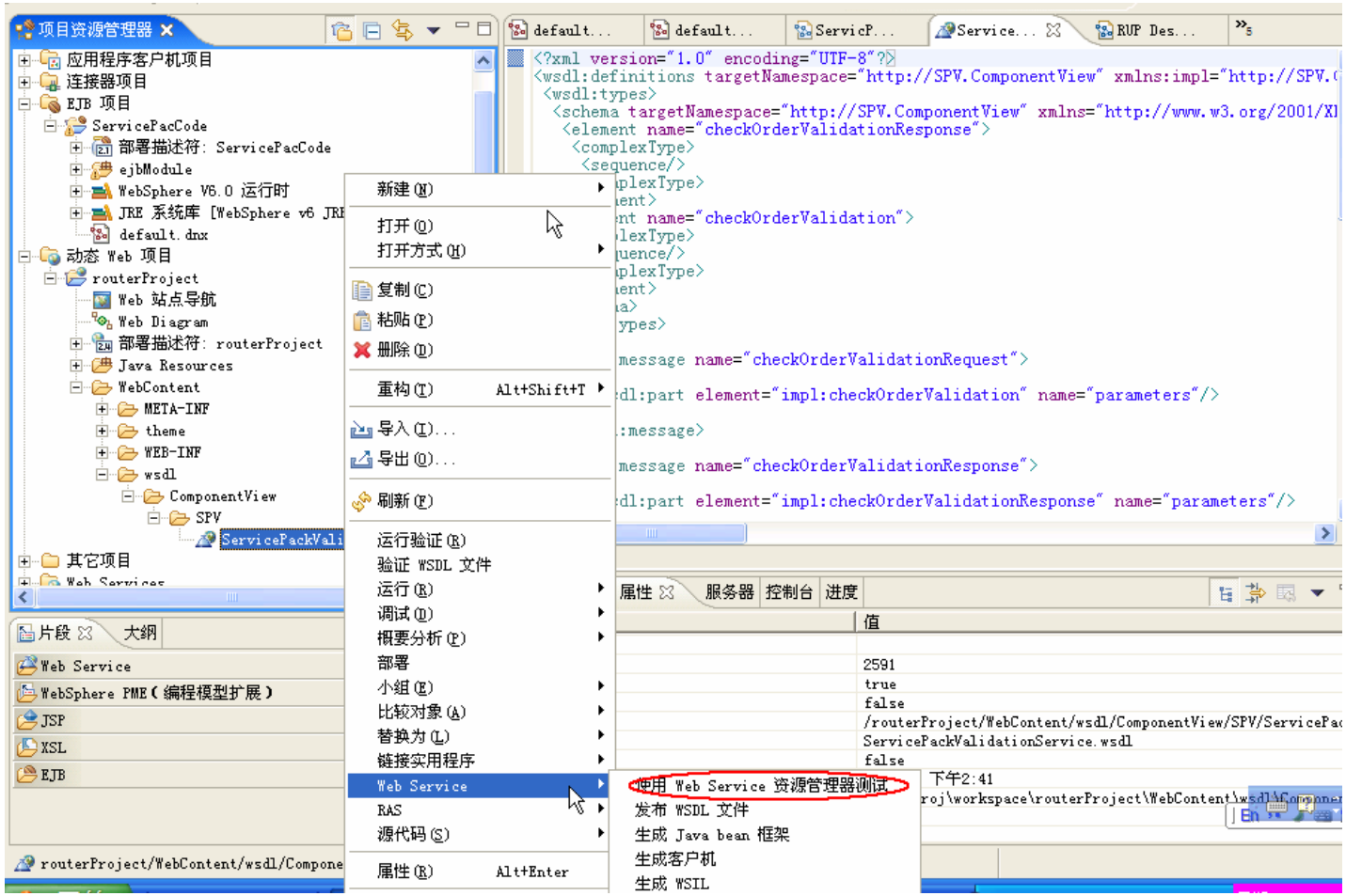
Bottom Up – 从EJB创建Web Service



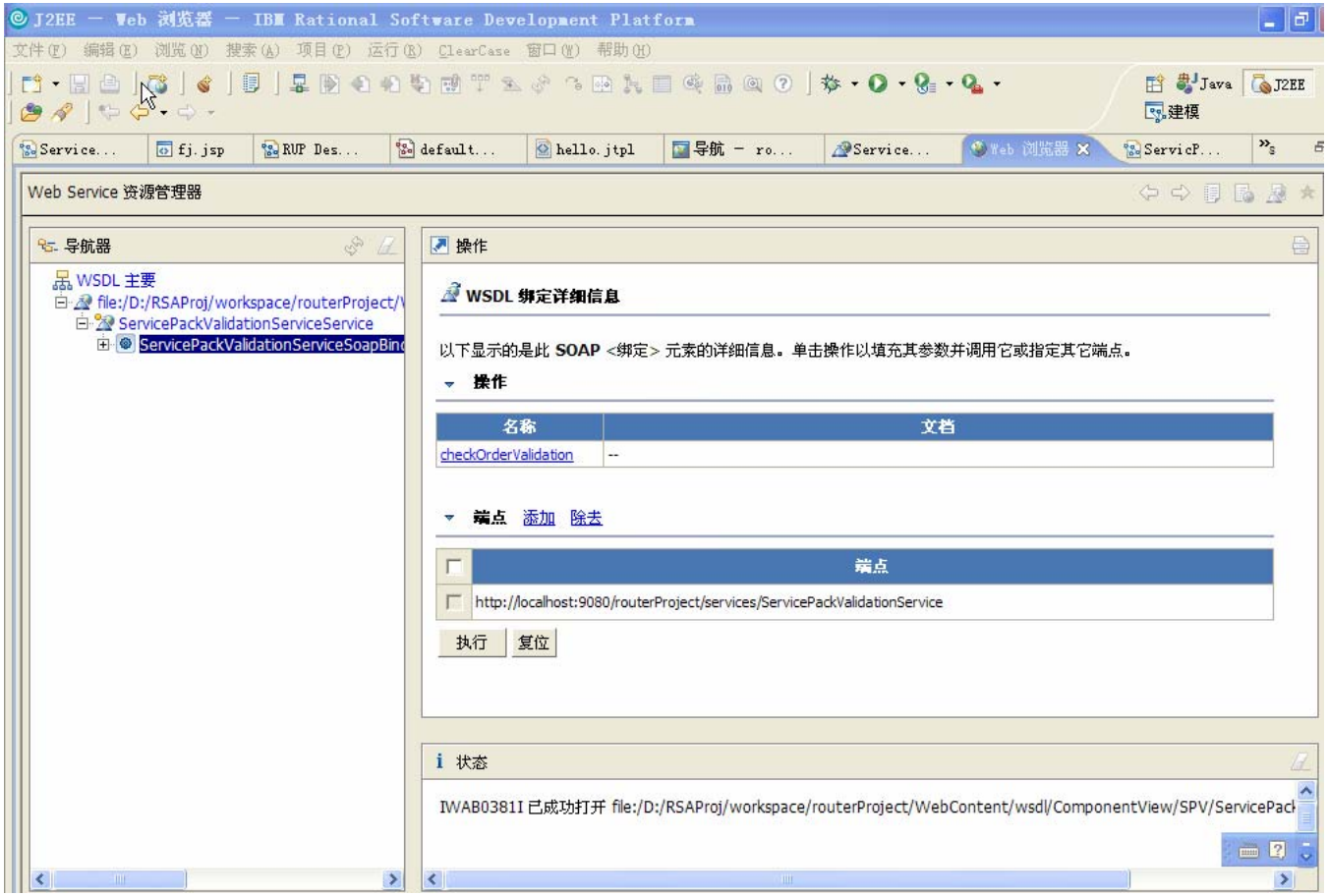
Top Down –从WSDL自动生成JAVA调用的代码



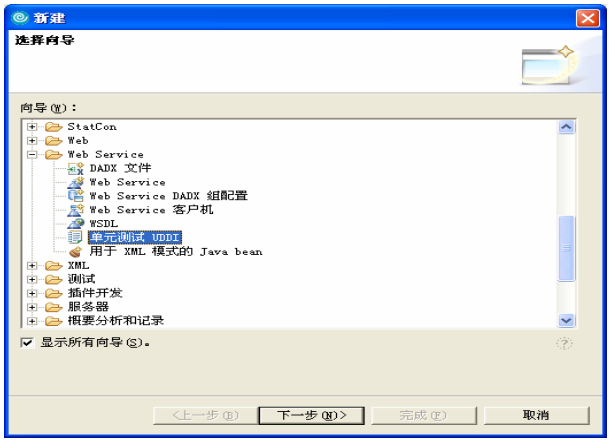
使用WSDL 浏览器进行Web Service执行测试

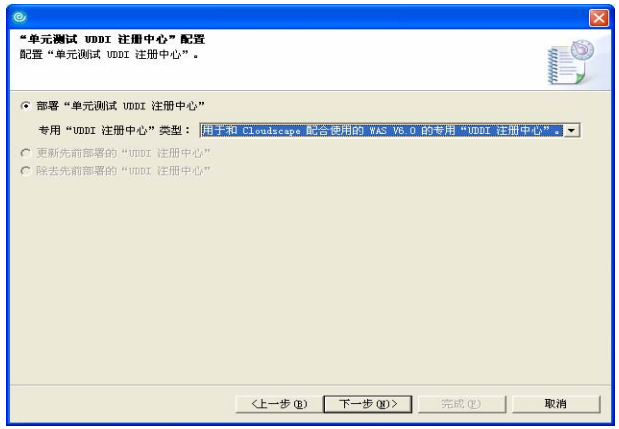


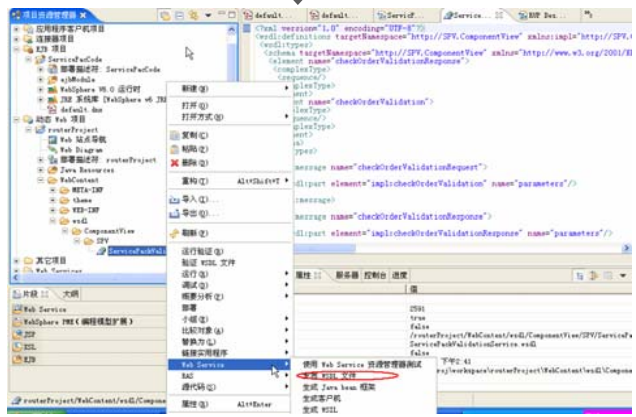
使用WSDL 浏览器进行Web Service执行测试

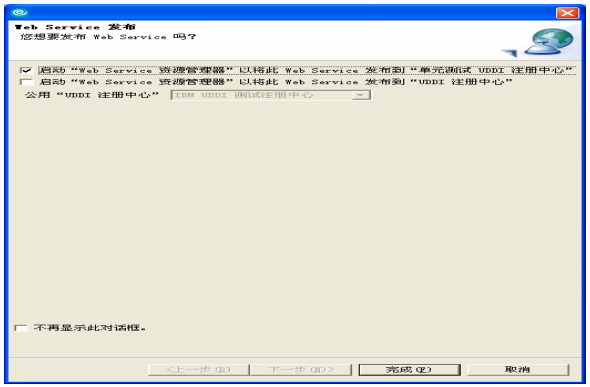


创建UDDI注册器并发布WSDL









内容

- 快速构建Web Service
- 快速构建JSF＋SDO

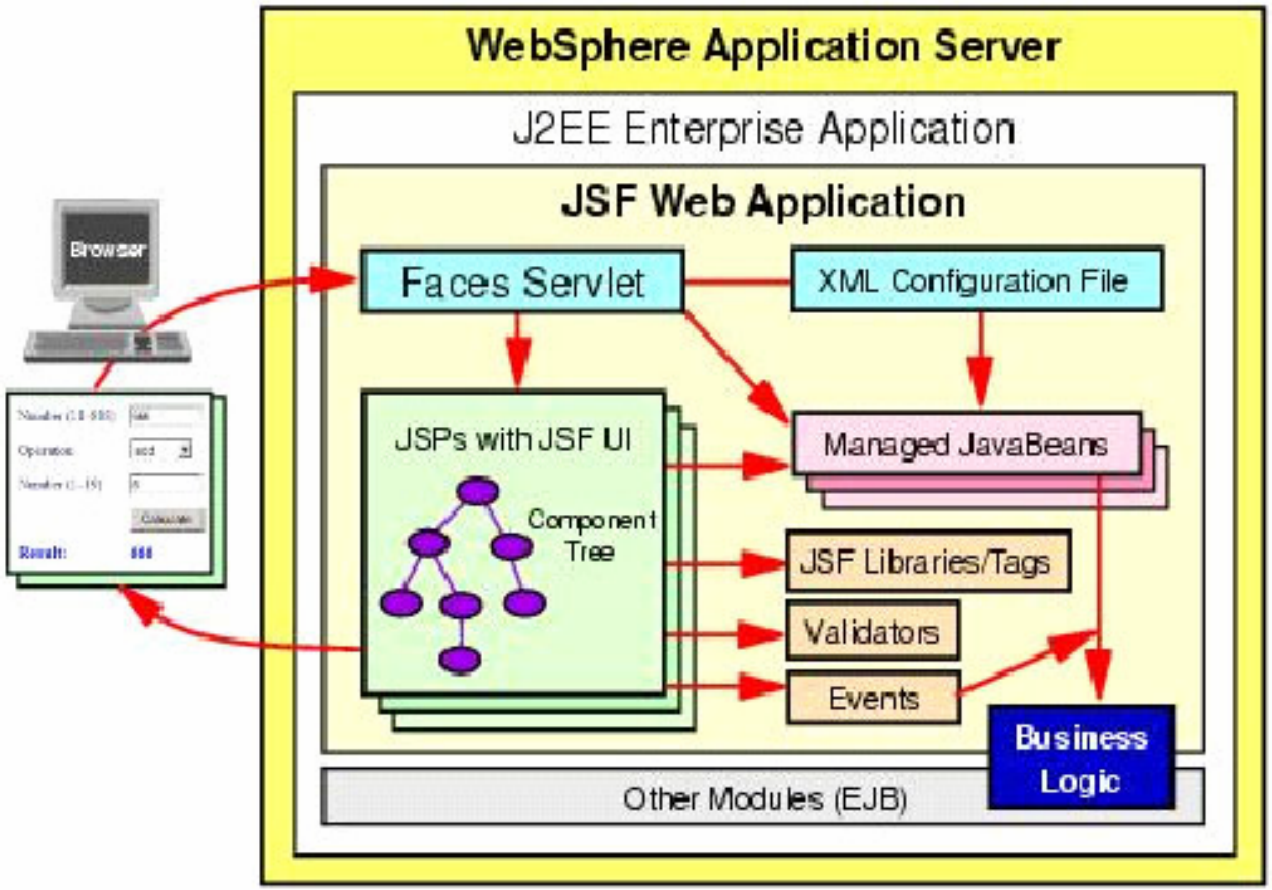
JSF简介

JSF为开发 Web应用用户界面提供了：

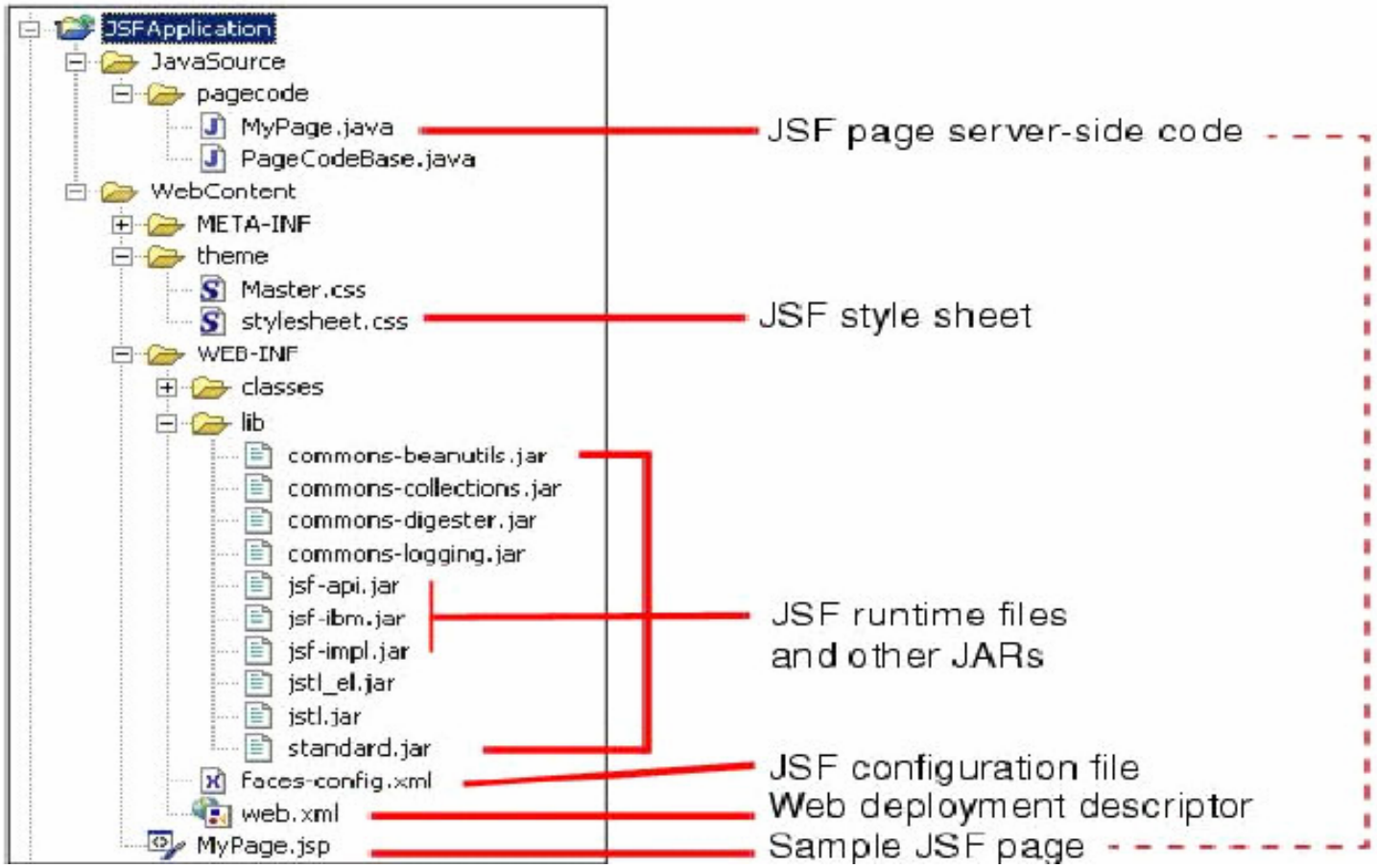
- 标准的编程接口
- 丰富可扩展的UI组件库
- 事件驱动模型等一套完整的Web应用框架

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

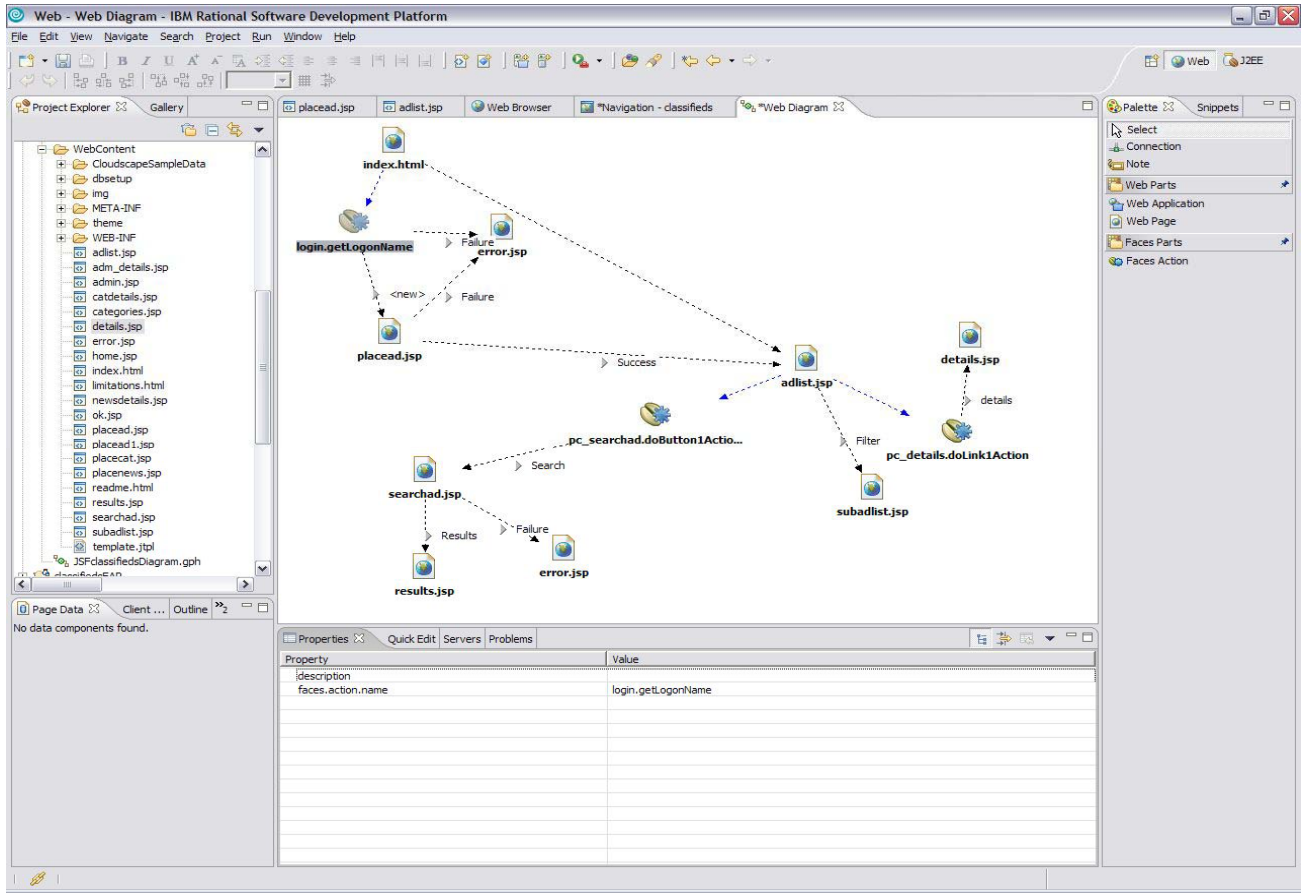

WebSphere Application Server
J2EE Enterprise Application
JSF Web Application
Browser
Faces Servlet
XML Configuration File
JSPs with JSF UI
Component Tree
Managed JavaBeans
JSF Libraries/Tags
Validators
Events
Business Logic
Other Modules (EJB)

RAD开发JSF－目录结构



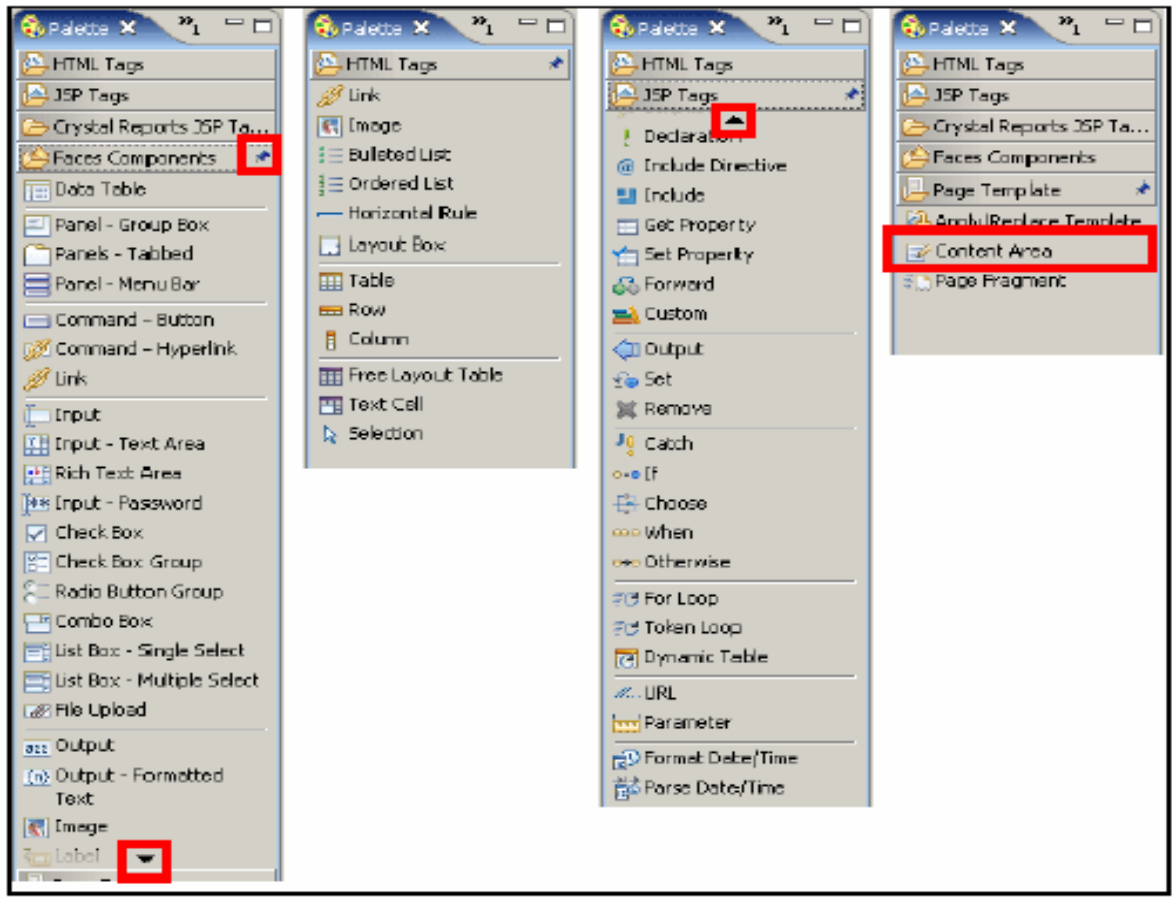
RAD开发JSF－Web Diagram

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



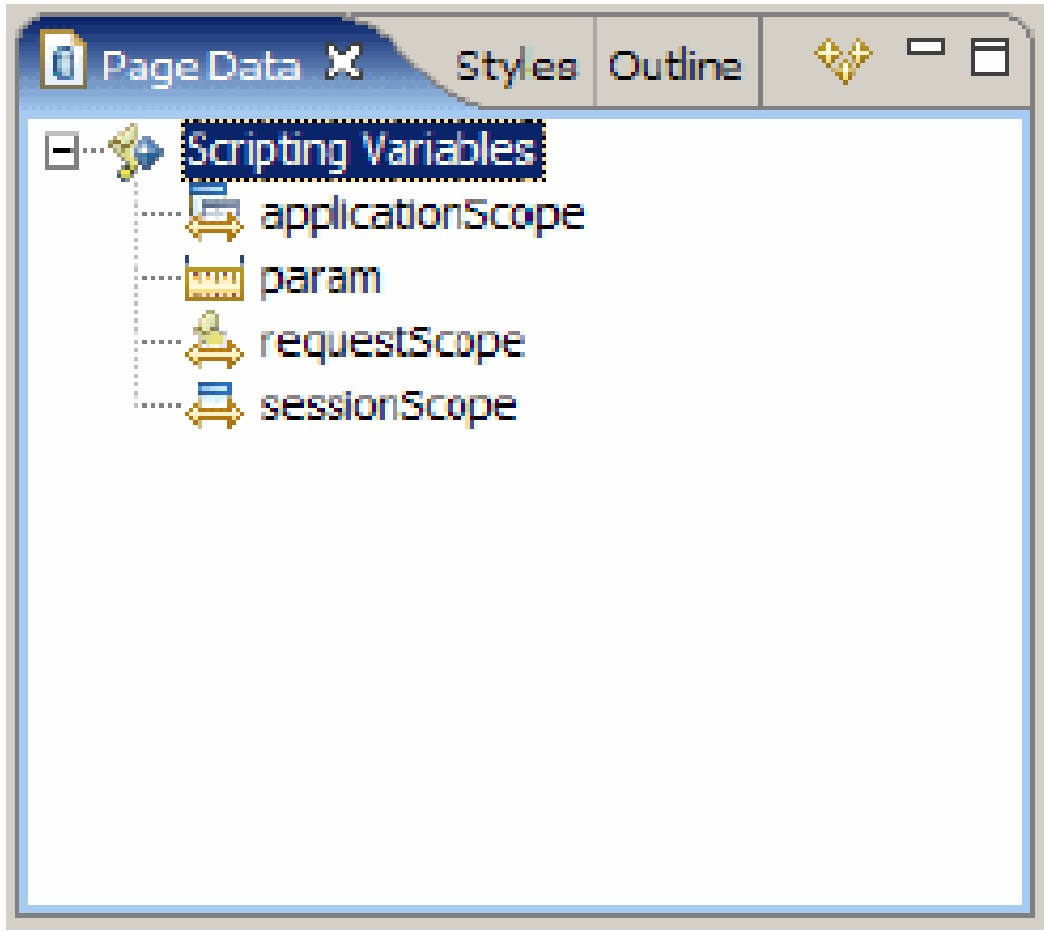
RAD开发JSF－Palette

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

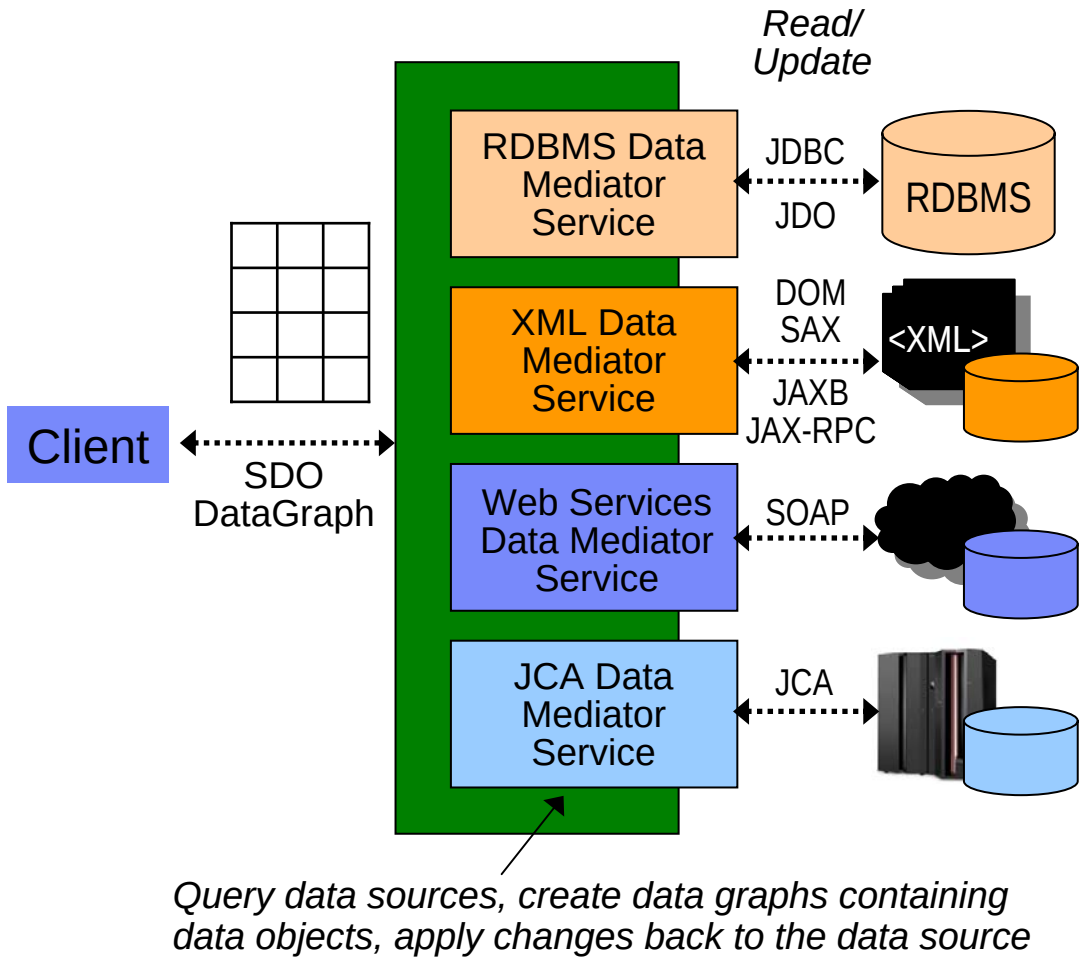
可以将各种数据源如Web Service, Database(SDO)和EJB Session Bean等加入到页面中



Service Data Objects (SDO)

Emerging Standard

- SDO是Java平台的数据编程架构和API
- 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
- 将逻辑代码和数据访问代码剥离
- 使应用程序、工具和框架更容易查询、读取、更新和检查数据
- 用于提高程序员的开发效率

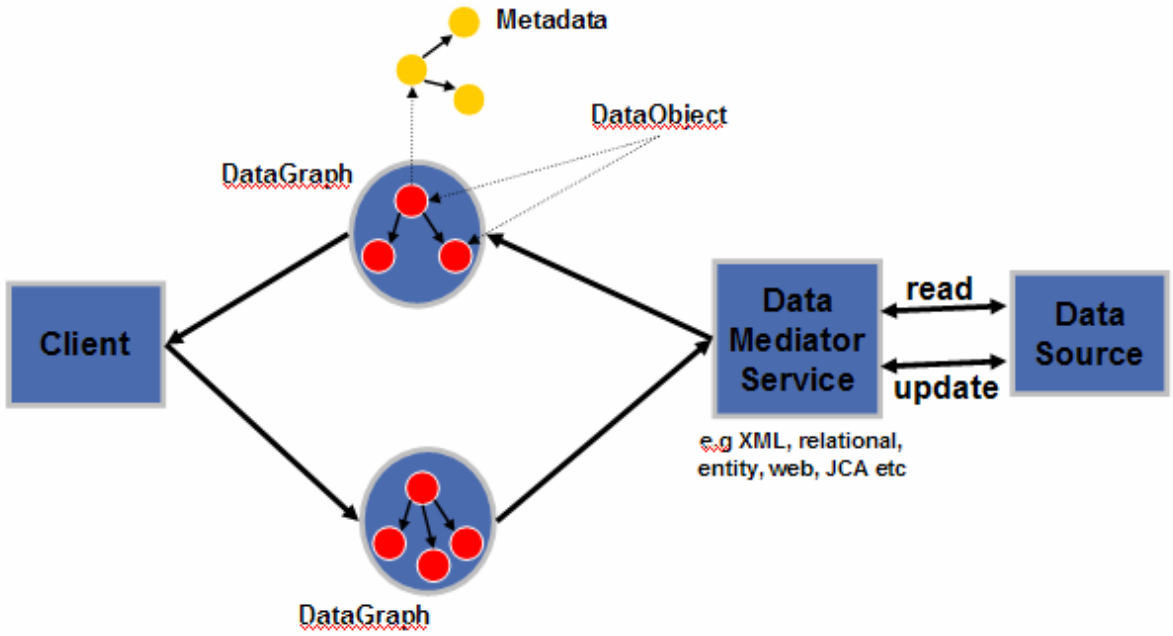


SDO是IBM和BEA共同发布的一项规范，目前正由JSR-235专家组进行标准化以通过JCP(Java标准化组织)的审核。

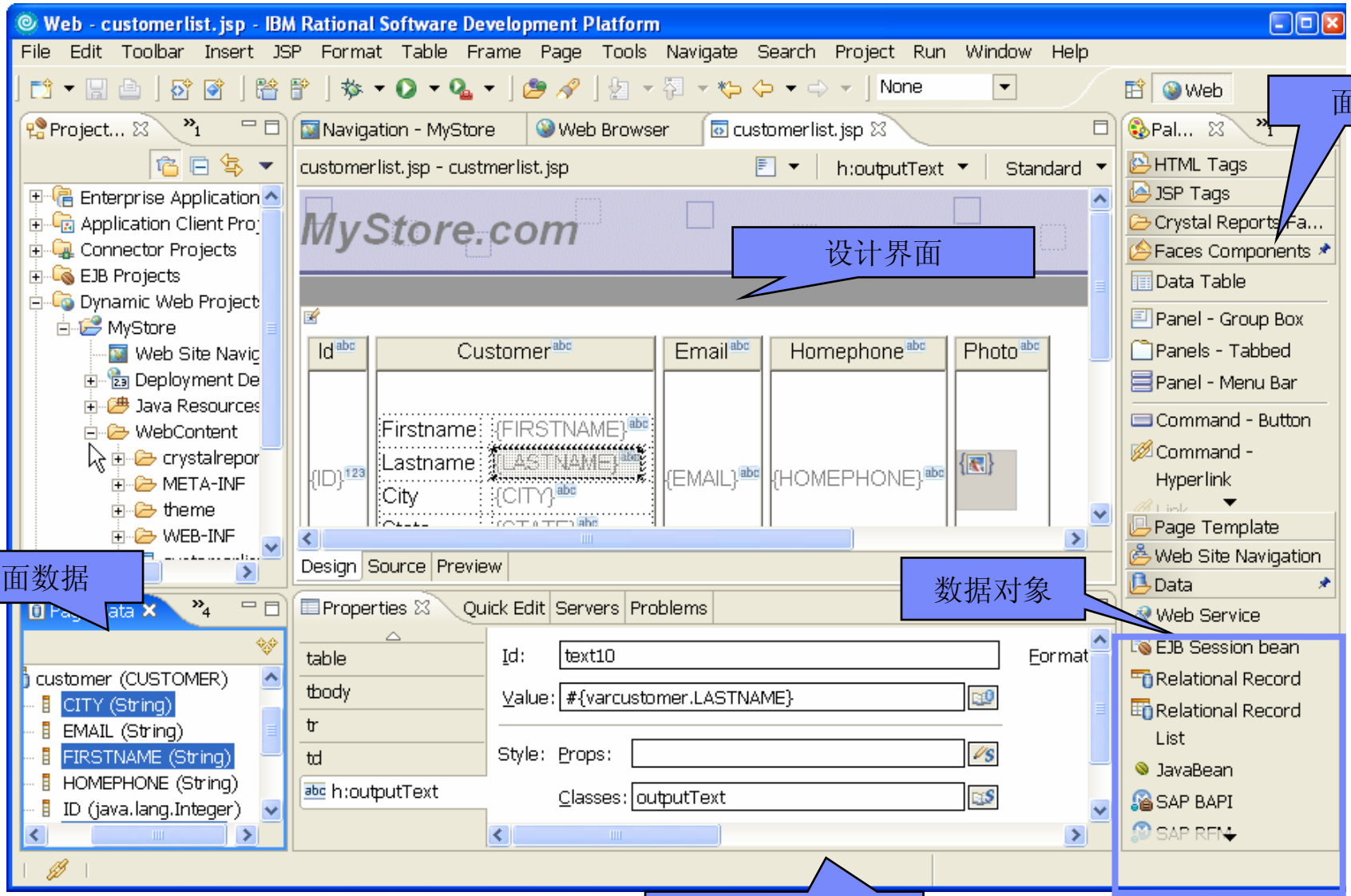
SDO的好处

SDO带来的好处：

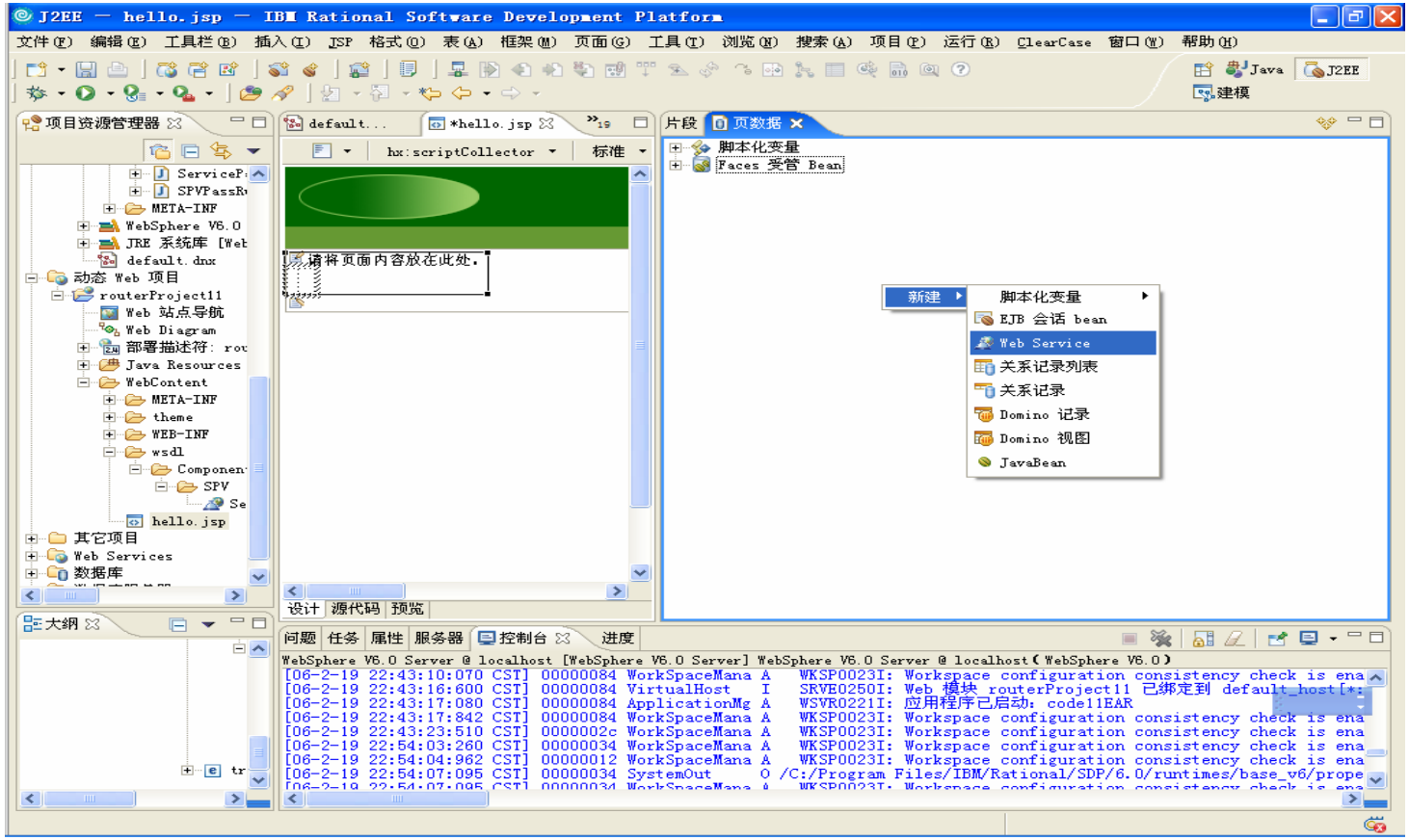
- 统一数据应用开发
- 简化J2EE应用层对数据层的访问
- 支持和集成各种数据源
- J2EE pattern和最佳实践的体现



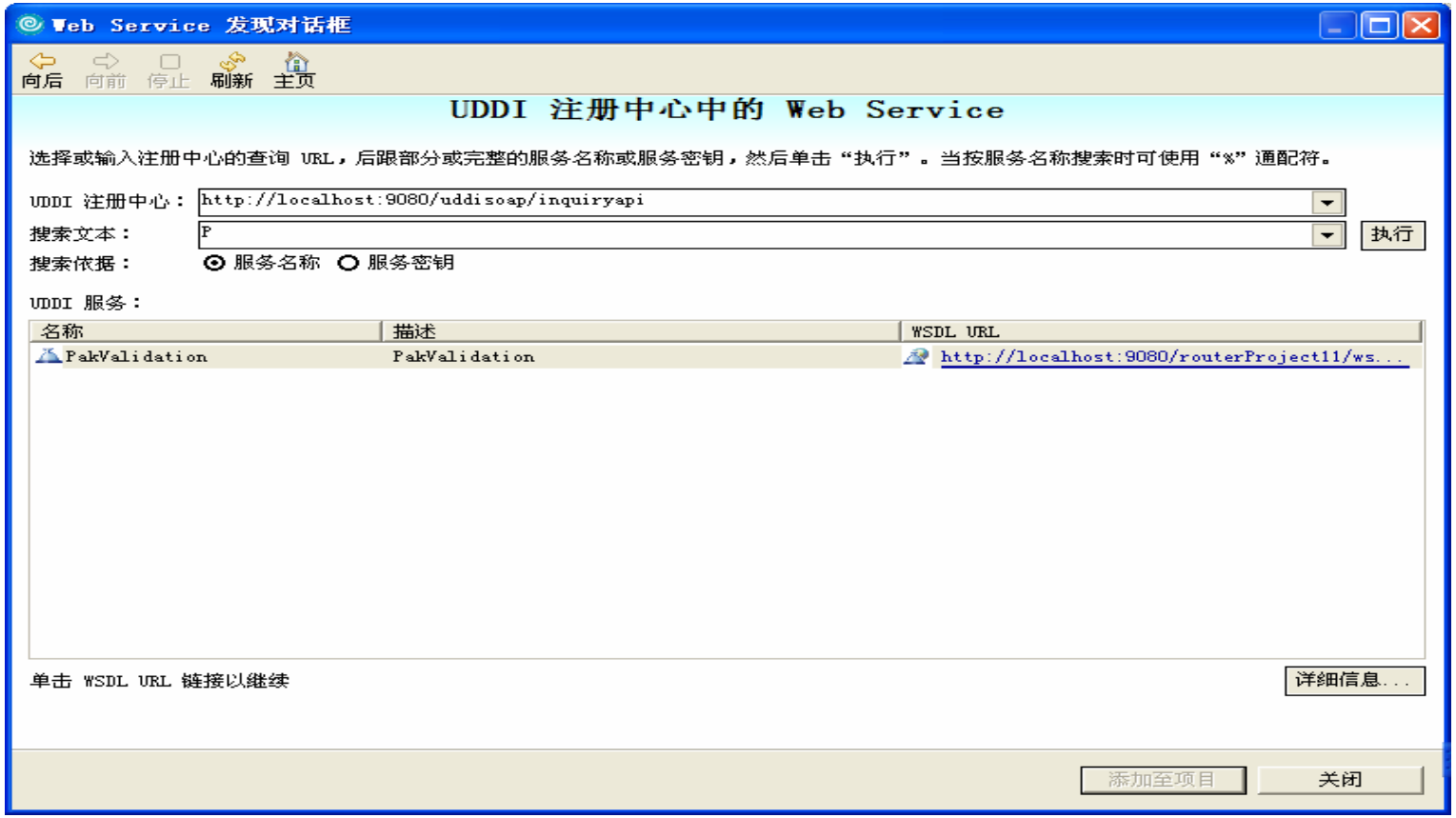
进行JSF/SDO的开发



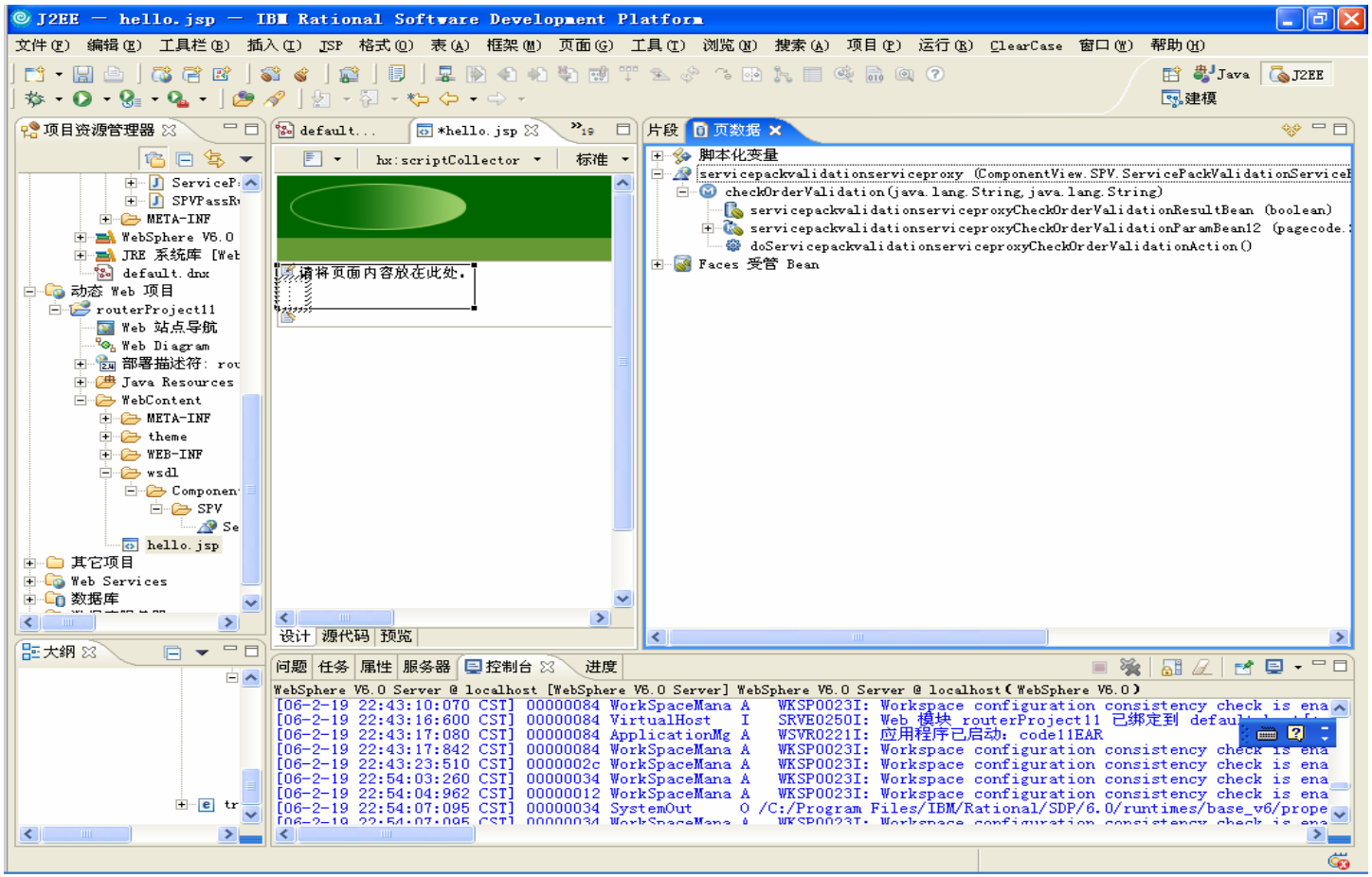
Demo9-使用SDO连接Web Service



使用SDO连接Web Service



使用SDO连接Web Service




Thank
You