

软件工程实践
Rational 用户组



uml.org.cn
UML 软件工程组织

基于RSA进行J2EE架构实践



邢波涛



Evolve by case

议程

- ➔ ■ 某银行ABC项目背景介绍
- J2EE概述
- RSA概述
- ABC项目非功能性需求
- ABC项目系统架构设计
- 利用RSA对ABC项目进行详细设计



某银行ABC项目背景介绍:

- 由于客户知识产权原因，不能公开，参见 培训课程
\\P1. ppt



议程

- 某银行ABC项目背景介绍
- ➔ ■ J2EE概述
- RSA概述
- ABC项目非功能性需求
- ABC项目系统架构设计
- 利用RSA对ABC项目进行详细设计

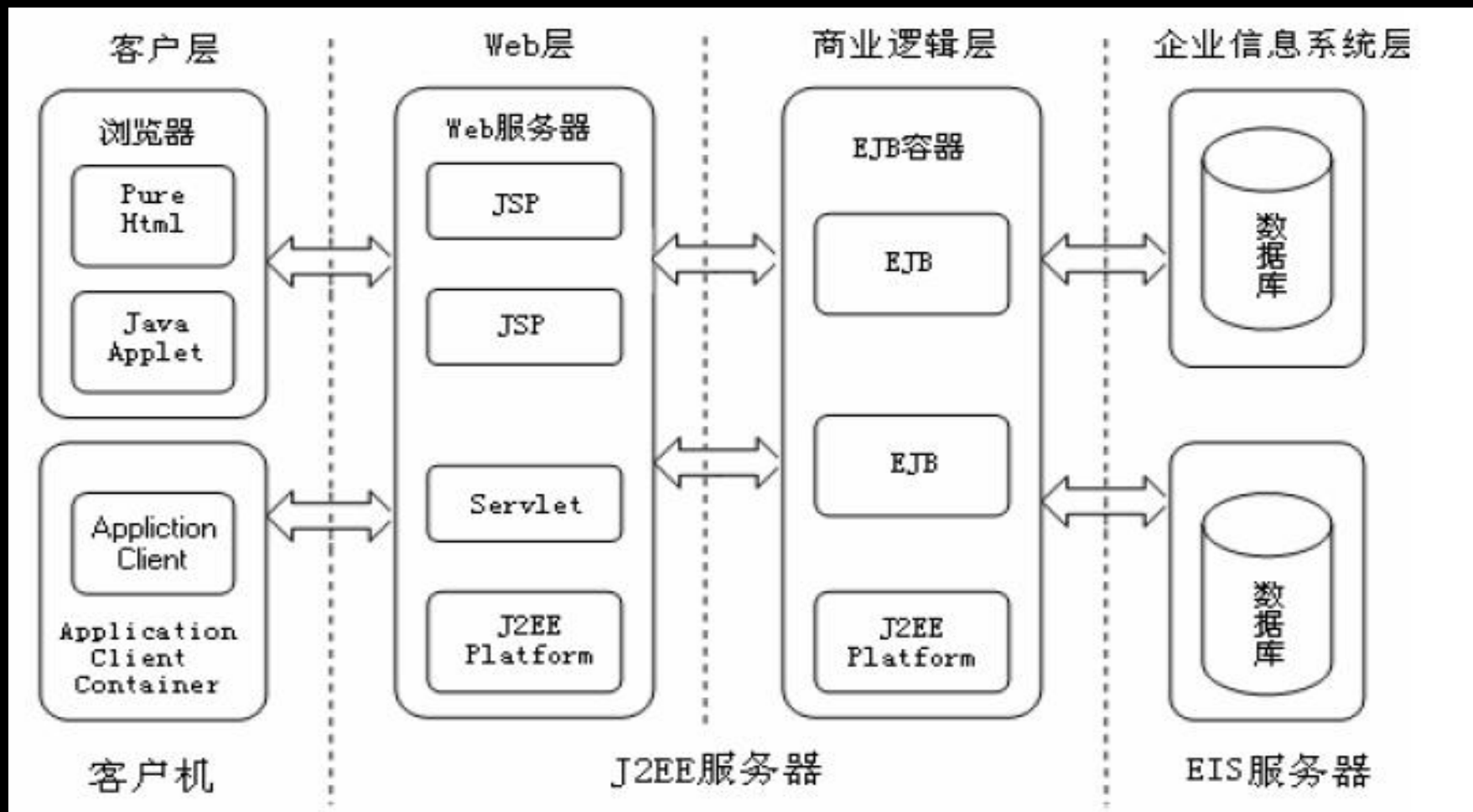


J2EE概述:

- 什么叫J2EE?
 - ▶ Java 2 Platform, Enterprise Edition
- J2EE是**分布式的多层**应用程序(两条主线):
 - ▶ J2EE平台使用多层分布式的应用模式;
 - ▶ 应用逻辑根据其功能分成多个组件, 各种不同的应用组件构成分布在不同的依赖于层的机器上;
 - ▶ 下面列出了位于不同层的组件:



J2EE概述:



J2EE概述:

- ▶ 运行在客户机上的客户层组件
- ▶ 运行在J2EE服务器上的Web层组件
- ▶ 运行在J2EE服务器上的逻辑层组件
- ▶ 运行在EIS服务器上的企业信息层（EIS）软件

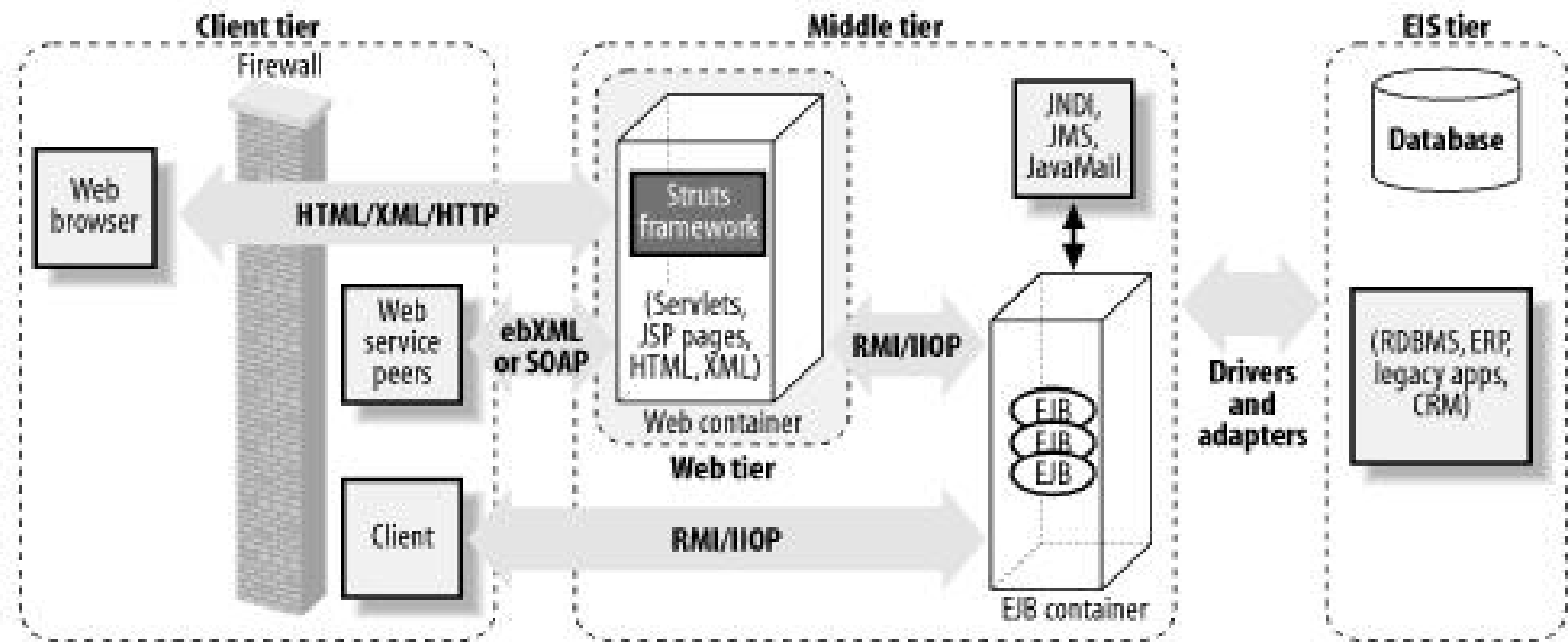
尽管J2EE应用程序可以由三层或四层或者N层构成，J2EE应用程序通常由三层构成，因为它们分布于三个不同的位置：客户，服务器，后台数据库服务器。通过这种方式运行的三层应用模式拓展了基于C / S 的两层模式。



J2EE概述:

- 一个典型的Struts应用是如何分多层的:
 - ▶ 从整个J2EE应用来说, Struts是位于Web层的

The Struts framework is used within the web tier



J2EE为什么要分多层:

■ 软件层的特征

由于每个软件都有自己的特点，因此不可能提供一个适合于所有软件的体系结构。总的说来，软件的层必须符合以下特征：

- ▶ 每个层由一组相关的类或者组件(如EJB)构成，共同完成特定的功能。
- ▶ 层与层之间存在着自上而下的关系，即上层组件会访问下层组件的API，下层组件不应该依赖上层组件。例如：WEB层依赖于商业逻辑层，商业逻辑层依赖于DAO层，DAO层依赖于数据库层。
- ▶ 每个层对上层公开API，但具体的实现细节对外透明。当某一层的实现发生改变，但是只要它的API不变，不会影响其他层的实现。



J2EE为什么要分多层:

出发点（背景）：把一个复杂的大系统拆分成多个小系统，拆分的过程也就是一个复杂——>简单的过程，每个层之间各司其职。

■ 多层带来的好处:

- ▶ 伸缩性:多层方式可以使企业级应用具有很强的伸缩性，它允许每层专注于特定的角色。例如，让Web服务器负责提供页面，应用服务器处理应用逻辑，而数据库服务器提供数据库服务。
- ▶ 可维护性:可维护性指的是当需求发生变化，只需要修改软件的某一部分。
- ▶ 可扩展性:可扩展性指的是在现有系统中增加新功能的难易程度。层数越少，越容易破坏现有的程序结构。
- ▶ 可重用性:可重用性指的是程序代码没有冗余，同一个功能能够满足多种需求。例如：商业逻辑层即可以满足GUI界面的表述层，也可以支持WEB界面的表述层。



J2EE为什么要分多层:

■ 多层带来的坏处:

- ▶ 层次之间联系过于紧密，客户层和后端的业务对象具有较强的依赖关系；
- ▶ （经常会有人讨论POJO/BO/VO/Struts 的FormBean在层之传递的问题）
- ▶ 在客户和服务器之间有多次方法调用，因而导致了Web性能方面的问题；
- ▶ 缺乏一定的客户访问机制，使得一些后台对象被随便访问。



J2EE 13种核心技术:

- JDBC?
- Java Server Pages?
 - Servlet
 - Java XML
 - RMI
- JavaBeans? Activation Framework
 - Java Messaging Service
 - JavaMail
 - JTS
 - JAVA事务 (JTA)
 - Java IDL/CORBA
 - JNDI
- Enterprise JavaBeans



J2EE核心模式介绍：

J2EE有两大类重要J2EE模式，一类由Sun Java Center管理，定义15种模式，在《Core J2EEMPatterns》书中发表，另一类是TheServerSide.com. 它发表了17种模式，最重要的见((EJB Design Patterns: Advanced Patterns, Process and Idioms))。

Sun Java Center模式类别中的15个模式与TheServerSide类别中的17个模式是相联系的，有些模式是相似的，提供同一问题的相似解决方案。下面我们只介绍Sun定义的15种J2EE核心设计模式。

Sun 15 种核心J2EE模式主要针对Web组件与业务逻辑 (EJB)层, 其组织如下：

- ▶ 表示层模式，用于Web 组件层；
- ▶ 业务层模式，用于组件与业务逻辑 (EJB)层；
- ▶ 集成层，用于连接EIS 层的系统。



最重要的J2EE设计模式:

■ 表示层（6类）：

▶ Intercepting Filter(截获过滤)

提供请求预处理和后处理的方案，定义灵活的体系结构，可以声明对截获请求和响应进行过滤。小服务过滤是这个模式的实现方法。

▶ Front Controller(前端控制器)

通过中央控制器提供请求的管理和处理。控制器取代通常发生在表式层的请求，从而取代模型视图控制器(MVC, Model View Controller)模式的控制器部分。前端控制器管理内容读取、安全性、视图管理和导航。

▶ View Helper(视图帮助器)

将负责表示(格式化输出—视图)的编程逻辑与其他(业务)逻辑分开。表示格式放在视图组件中，可能包括多个子组件，组成复杂视图。业务逻辑代码放在帮助器组件中。帮助器组件的典型功能是内容读取、验证与适配。帮助器组件可以用Business Delegate模式访问业务服务。



最重要的J2EE设计模式:

▶ Composite View(复合视图)

是从原子子组件创建累计表示(视图)的灵活方案。表示的体系结构可以方便地组织基本视图组件,使表示很灵活,还可以进行其他工作包括个性化与定制。

▶ Service-to-Worker(服务/工人)

由Dispatcher组件与Front Controller和View Helper模式组合而成,先进行请求处理再进行视图处理,适合大型应用程序,类似于Dispatcher View模式。

▶ Dispatcher View(派遣视图)

类似于Service to Worker模式,也是由Dispatcher组件Front Controller和View Helper模式组合而成。但与Service-to-Worker模式不同的是,这个模式在进行视图处理的期间进行请求处理,因此更适合小型应用程序。



最重要的J2EE设计模式:

■ 业务层模式(7个):

▶ Business Delegate(业务代理)

减少层之间的藕合，特别是表示层与业务逻辑层之间。它提供门户代理，可以作为业务逻辑层和EIS层服务的通用入口。代理还可以缓存远程方法调用，从而提高性能。这个模式可以和Service Locator模式组合。

▶ Value Object(数值对象)

解决层之间交换数据及相关远程开销的问题。这个模式收集序列化对象中需要的所有数据，用单个远程方法调用在层之间传输。这个模式通常是层之间交换数据和进行通信的最佳方法，可以成功地减少远程方法调用，这在分布式系统中成本很高。

▶ Session Fagade(这是J2EE模式中最最重要的一个)

隐藏业务组件的细节和集中工作流程，提供客户机的粗粒接口，减少远程方法调用开销，也适合声明式事务和安全管理。这个模式通常和其他模式组合，如ServiceLocator, Value Object, Value Object Assembler, Value List Handler, DAO。



最重要的J2EE设计模式:

- ▶ Composite Entity(复合实体)

解决EJB 1.1实体Bean的远程接口开销, 提供设计粗粒实体Bean的方法, 将相依赖对象合并为一个实体Bean, 用于EJB 1.1持久性模型, 在EJB 2.0持久性模型、本地接口和管理关系中已经过。

- ▶ Value Object Assembler(数值对象汇编器)

可以灵活地复合不同来源的Value Object. 包括EJB, DAO或Java对象(POJO)。这个模式创建要传递到客户机上的数据。这个模式与Value Object模式相联系。

- ▶ Value List Handler(数值清单处理器)

提供杏询执行和结果处理的良好方案, 还可以缓存结果, 进一步提高性能。这个枝式基于前面提到的23种设计模式的Iterator模式。

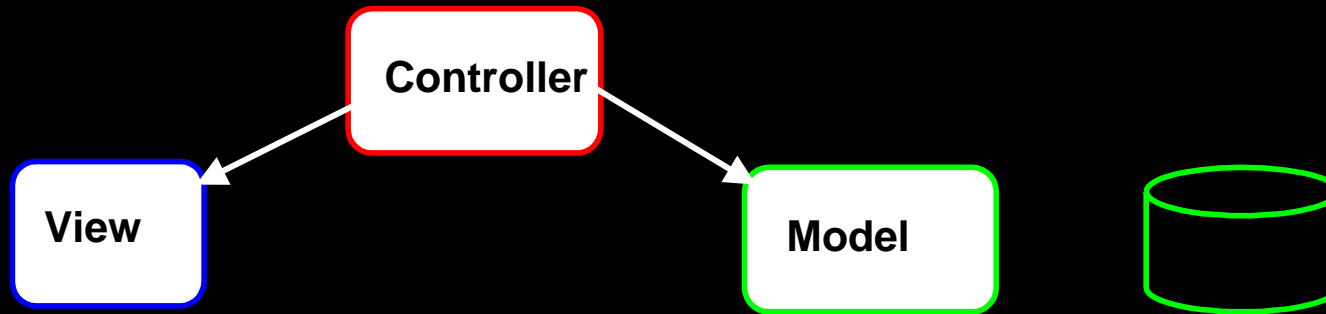


最重要的J2EE设计模式:

- ▶ Service Locator (服务定位器)
可以查找、创建与定位服务工厂，包装其细节。多个客户机可以使用这个对象，从而减少复杂性，提供单个控制点，通过缓存提高性能。
- 集成层模式 (2个):
 - ▶ Data Access Object (数据访问对象)
可以灵活而透明地访问数据，抽象数据源和隐藏EIS表示层细节。优点是业务与EIS层之间得到松耦合。这个模式也用于实体Bean的Bean管理持久性。
 - ▶ Service Activator (服务激活器)
可以异步处理同步EJB组件，特别是会话Bean。这个模式提供类似于EJB 2.0规范中消息驱动Bean的功能。这个模式的优点也适用于EJB 1.1。



“Model-View-Controller” (MVC)

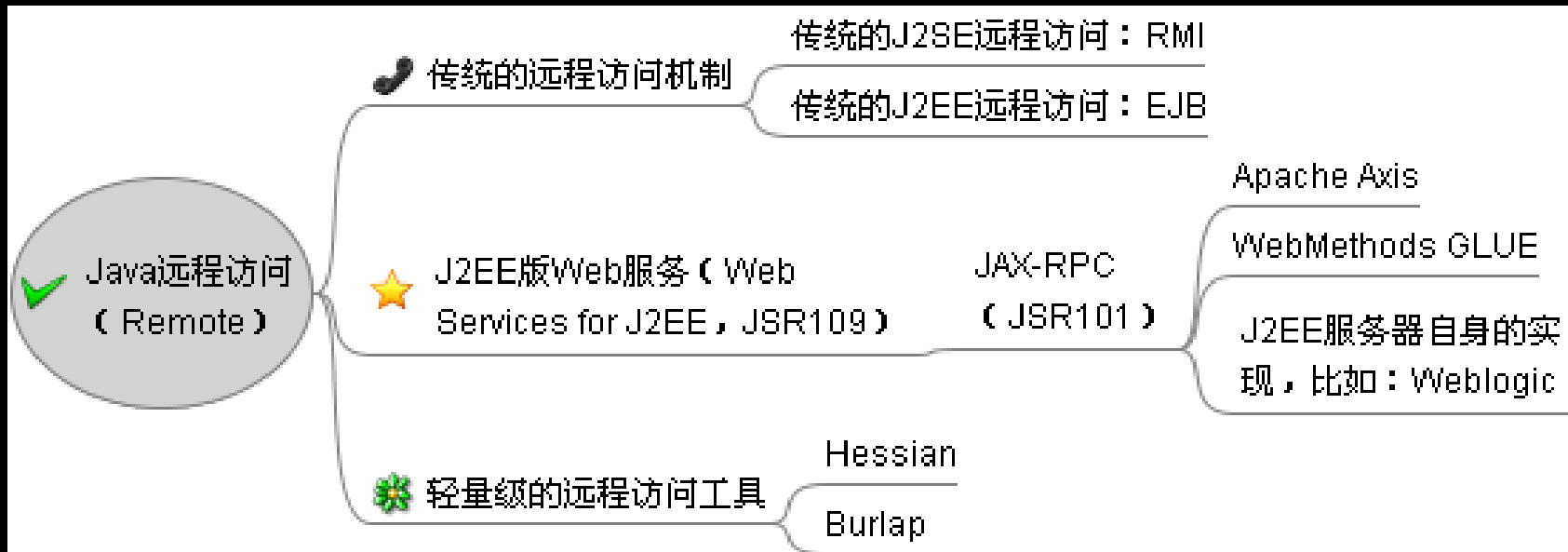


- 表示与逻辑分开
- View - 表示: 显示输入输出数据 (可以是 HTML, 可以是一个 windows 应用)
- Model - 业务逻辑和数据: 基于输入和定义的业务流程执行计算 或其它操作
- Controller - 协调view 和 model, 在它们之间交换数据



J2EE另外一条主线,分布式:

- J2EE另外一条主线: 分布式
 - ▶ 首先看一下分布式应用在J2EE架构下实现途径



企业级应用, 就要兼顾很多问题了, 事务, 并发, 安全, 缓存... 当然RMI, 和 webservice都可以实现, 但是却具备不了企业级应用必须的东西, 如果单单是为了实现分布式, 那上面列举Hessian, Burlap, XmlRpc..... 这些东西都可以实现, 但是如何兼顾事务, 并发, 安全, 缓存...这才是我们必须考虑的问题, 而EJB恰好解决了这个问题。



议程

- 某银行ABC项目背景介绍
- J2EE概述
- ➔ ■ RSA概述
 - ABC项目非功能性需求
 - ABC项目系统架构设计
 - 利用RSA对ABC项目进行详细设计



RSA概述:

- 什么是Rational Software Architect?
 - ▶ IBM® Rational Software Architect -- IBM软件开发平台的一部分 - 是IBM在2003年二月并购Rational以来, 首次发布的Rational产品。改进过的软件开发平台在集成和易用性上达到一个新的层次。
 - ▶ IBM Rational Software Architect (RSA) 是一套设计与开发工具, 它构建在开放的、可扩展的Eclipse3.0平台之上, 实现了多项行业最新标准, 提供了灵活的插件扩展机制。借助UML2.0技术, 它**实现了模型驱动的开发模式**, 可以帮助开发团队创建更加强壮的软件架构。同时, RSA作为IBM Rational业务驱动软件开发平台的核心构件, 提供了与需求管理工具、测试工具、配置和变更管理工具和项目管理工具的完美集成, 从而真正实现了企业内部的核心软件开发流程、开发平台和软件生产线。



RSA概述:

- 软件开发平台包含了好几个IBM Rational产品：
 - ▶ IBM Rational Portfolio Manager: 是一个围绕整个项目生命周期的项目管理工具，Rational Portfolio Manager有助于对项目组合投资提供实时的可见性。它允许管理者作出正确的决定。这个产品也可以用来增进不同开发团队间的协作。
 - ▶ IBM Rational Application Developer and Rational Web Developer: 这分别是 IBM WebSphere Studio Application Developer和 IBM WebSphere Studio Site Developer 集成开发环境（IDE）工具的另一个商标。这些新产品现在是IBM Rational Software Modeler 和 IBM Rational Software Architect 的基础。
 - ▶ IBM Rational Software Modeler: Rational Software Modeler 是一个基于统一建模语言（UML）2.0的工具，它允许架构设计师和分析师们创建系统的不同视图。这个工具一方面能很好的促进架构师和设计师的沟通，另一方面也能促进架构师和开发团队的沟通。
 - ▶ IBM Rational Software Architect: 这个工具允许架构师设计和维护应用程序的架构。Rational Software Architect 是本文的焦点。



RSA概述:

- ▶ IBM Rational Manual Tester: 这个工具允许分析人员和测试人员去创建和执行测试用例。Rational Manual Tester 是针对单元测试的。
- ▶ IBM Rational Functional Tester: 这个工具可以用来把功能测试和回归测试自动化。
- ▶ IBM Rational Performance Tester: 这个工具可以对你的应用程序自动进行增加负载和测试性能。

因为 Rational Software Architect 是 Rational Application Developer, Rational Web Developer和 Rational Software Modeler 的父集, 所以当你下载 Rational Software Architect, 你就获得了所有这些工具。

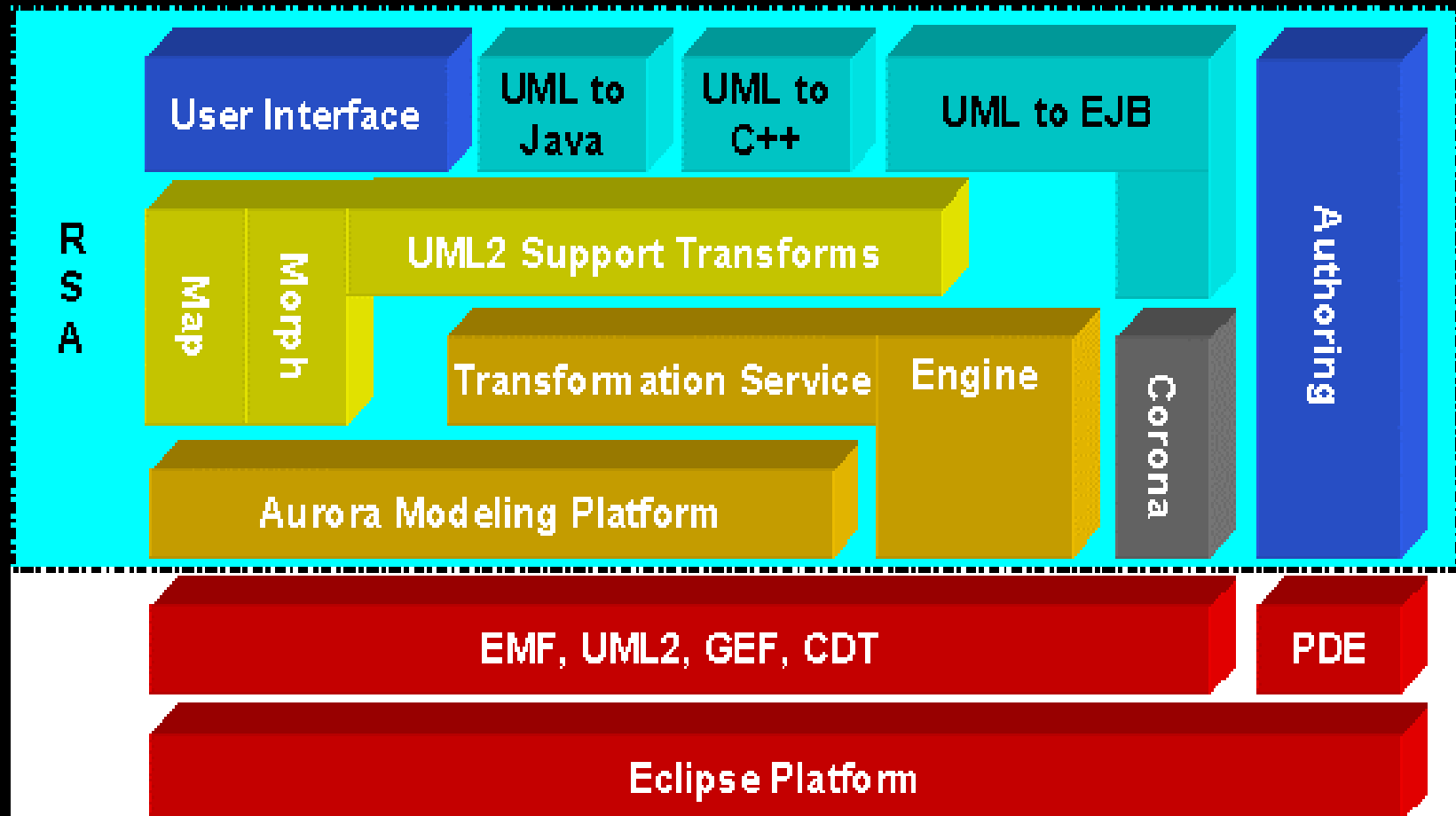


使用RSA实现MDA中的模型转换：

- IBM在RSA中缺省提供了UML到Java、UML到C++、UML到EJB的模型转化实现，其整个实现架构基于Eclipse 3.0和Eclipse Model Framework (EMF)、UML2、Graphical Editing Framework (GEF)和插件开发环境 (PDE)，为模型驱动软件开发 (MDA) 的模型转换方面提供了四个方面的工具支持：
 - ▶ 转换服务 (Transformation Service)：支持转换的交换和发现功能，为其它转换工具支持提供基础。
 - ▶ 转换引擎 (Transformation Engine)：为层次化的转换源提供遍历引擎，尤其对UML模型有较好支持。
 - ▶ 转换应用的GUI界面支持：使用户能够浏览、查找并应用各种转换。
 - ▶ 制作转换的GUI界面支持：为转化制作中的创建、打包和维护转换的各种任务，提供很多自动化支持。

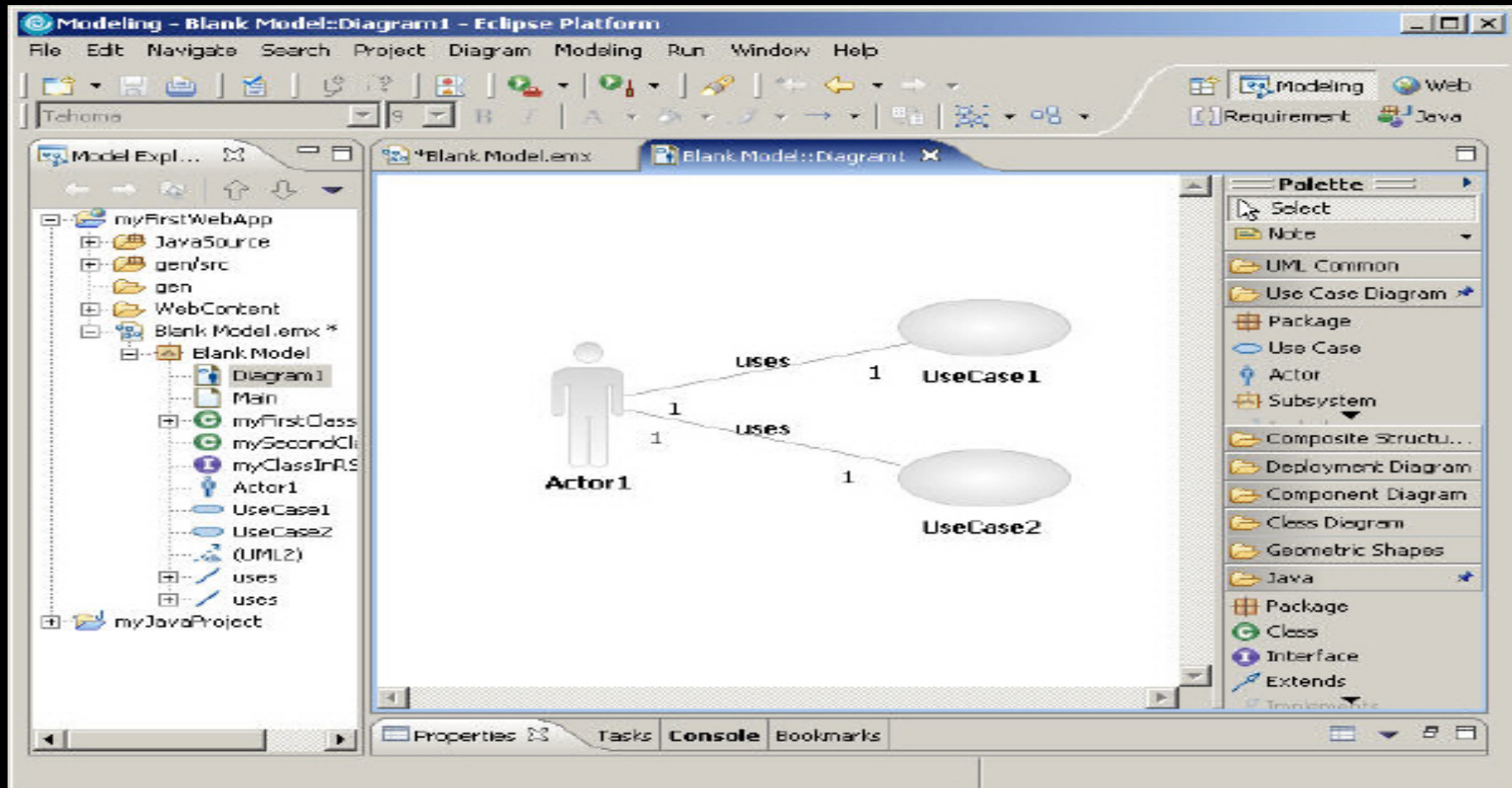


使用RSA实现MDA中的模型转换：



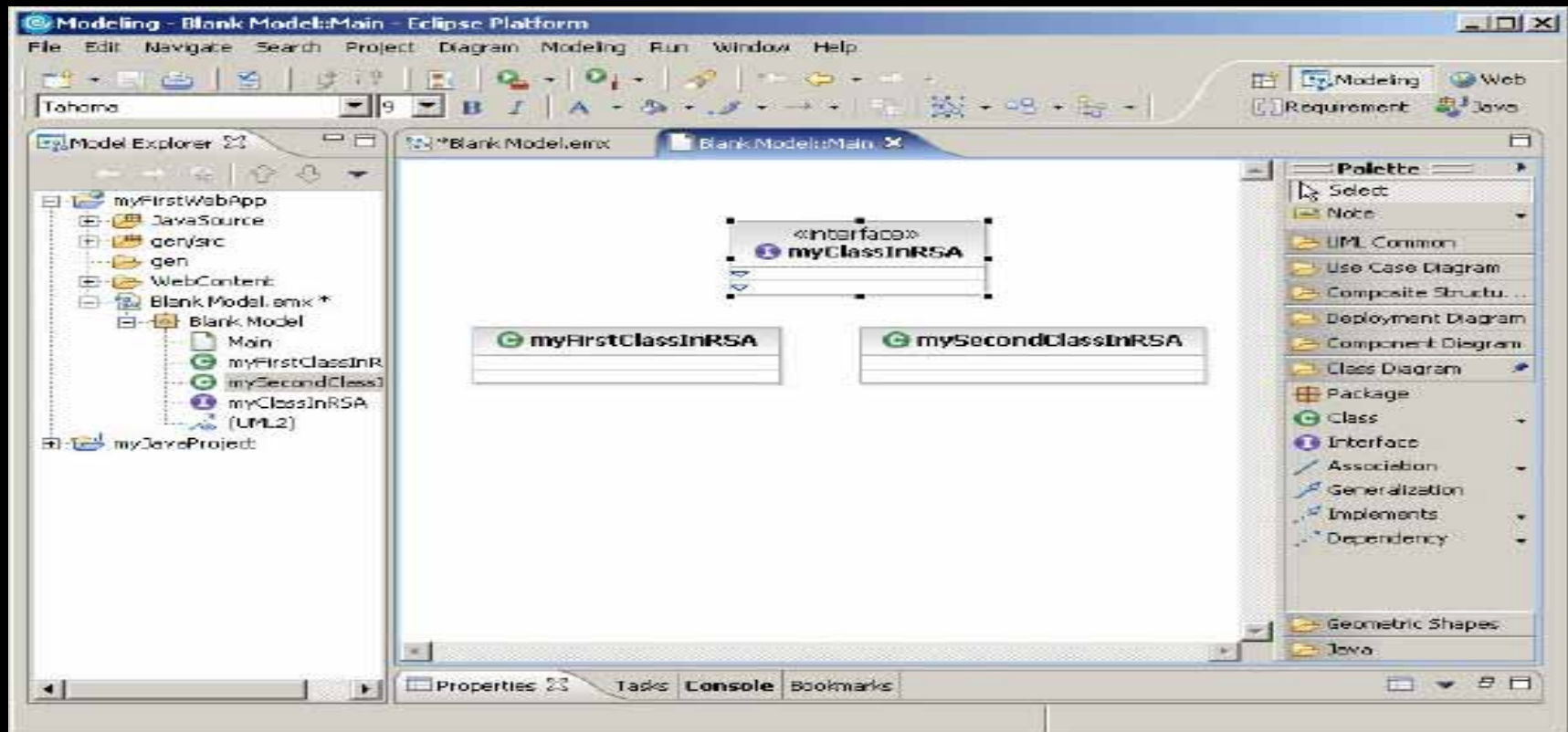
Rational Software Architect 进行UML建模：

- 创建一个 UML 用例图：



Rational Software Architect 进行UML建模：

- 创建一个 UML 类图：



Rational Software Architect 进行UML建模：

- UML 到编码

我们可以从刚刚创建的 UML 类图中生成代码。

- 代码到 UML

让我们反过来做一下：用你的代码生成一个 UML 模型。这个过程实际上相当容易。只是创建一个空白的类图。从 Java 项目中把代码拖到这个空白的图就行了。UML 类图会被自动生成。

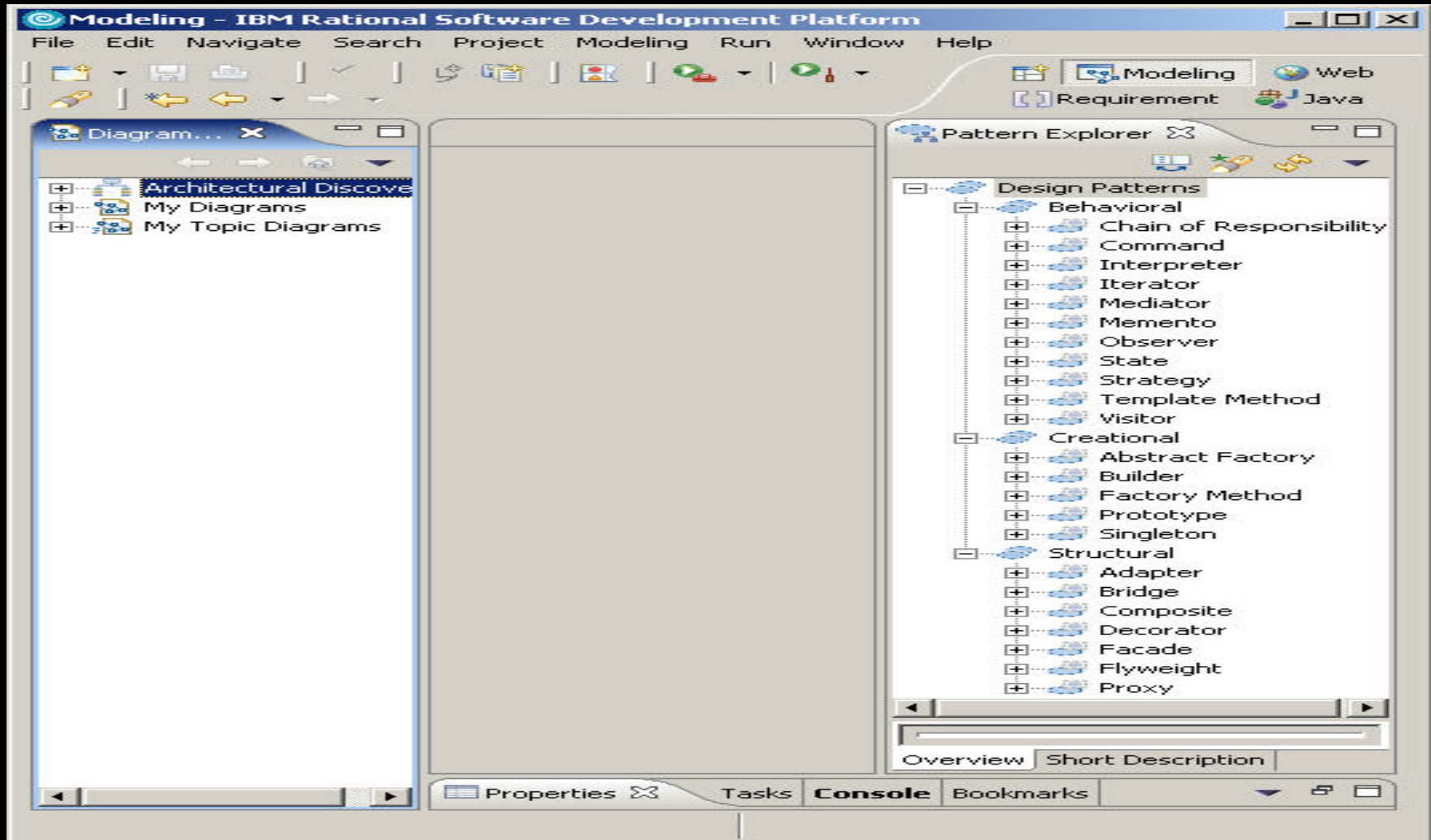


Rational Software Architect 中的设计模式

- Rational Software Architect 允许架构师设计并维护应用程序体系结构。
- Rational Software Architect 包含所有 23 个 Gang of Four 模式。（但是，您还可以创建并添加自己的模式）这些模式分成三个类别：
 - ▶ 创建型，处理对象或类的创建
 - ▶ 行为型，处理类或对象如何交互及分配职责
 - ▶ 结构型，处理类或对象的组合



Rational Software Architect 中的设计模式



议程

- 某银行ABC项目背景介绍
- J2EE概述
- RSA概述
- ➔ ■ ABC项目非功能性需求
- ABC项目系统架构设计
- 利用RSA对ABC项目进行详细设计



ABC项目非功能性需求：

- 以WEB为基础
- 由于银行柜员习惯于手工键盘操作，而不是鼠标，要求系统用户的界面操作应支持完全键盘输入完成。
- 表单内容需循环输入，不得利用鼠标跳跃式输入
- 表单字段输入后，立即检查正确性
- 支持键盘热键



议程

- 某银行ABC项目背景介绍
- J2EE概述
- RSA概述
- ABC项目非功能性需求
- ➔ ■ ABC项目系统架构设计
- 利用RSA对ABC项目进行详细设计



ABC项目系统架构设计

- ABC项目在进行架构设计时重点考虑因素：
 - ▶ WEB应用：基于WEB方式
 - ▶ 分布式应用：支持分布式
 - ▶ 用户系统操作习惯：键盘
- 现有技术参考和取舍：
 - ▶ Web层：JSP+XMLHTTP vs RichClient(Flex/Laszlo/SWT/Swing)
 - ▶ 商业逻辑层：EJB(Session Bean)
 - ▶ 持久层：Hiberbate vs EntityBean
 - ▶ 数据库：DB2



ABC项目系统架构设计:

由于客户知识产权原因，不能公开，参见 培训课程\P2.ppt



议程

- 某银行ABC项目背景介绍
- J2EE概述
- RSA概述
- ABC项目非功能性需求
- ABC项目系统架构设计
- ➔ ■ 利用RSA对ABC项目进行详细设计



利用RSA对ABC项目进行详细设计：

- RSA设计的ABC项目某交易序列图
- RSA设计的ABC项目类图

由于客户知识产权原因，不能公开，参见 培训课程\P3. ppt

