



Rational. software

软件架构管理 软件交付平台介绍手册(一)

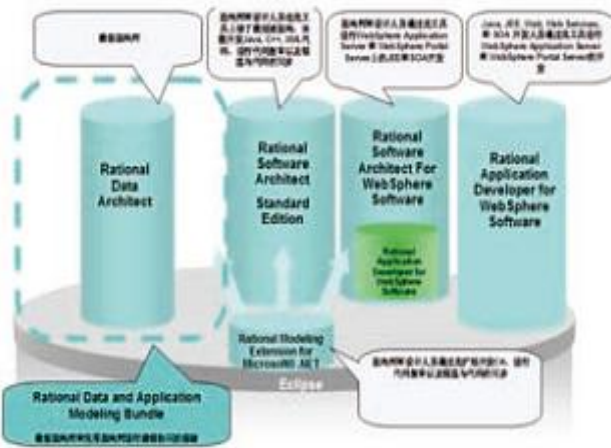


要想让软件更好地为业务服务，至关重要的一步就是实施一种能够对软件项目生命周期全过程进行管理的软件交付流程及支撑平台。

1 IBM Rational架构管理方案

在《软件开发平台整体介绍》文档中，介绍了IBM Rational软件交付平台主要包括的几个基本服务组件(如图一)：过程和组合管理，需求定义和管理，架构管理，质量管理和配置，变更和发布管理。在本文中，主要介绍架构管理部分(见图二)。

IBM Rational的核心能力 What we offer to help you get there



图二

基于Eclipse平台，开放而高效

常见问题：根据开发团队的不同角色采用不同的开发工具。工具与工具之间没有集成，团队成员的沟通必须通过频繁的数据导入导出操作来实现或通过频繁地切换工具，效率低下。

常见问题：开发团队使用封闭的，非开放的工具。当用户需要额外功能，而工具提供商无法为用户定制的时候，用户希望可自行扩展，但

是无法实现，因为工具是封闭的，不提供相应的接口。Rational新一代的设计和构建工具解决了以上问题。它们都是基于Eclipse的通用开发环境，这些工具包括：

IBM Rational Data Architect

IBM Rational Software Architect Standard Edition

IBM Rational Software Architect For WebSphere Software

IBM Rational Application Developer

所有的这些产品都有着相同的用户界面，我们称之为工作台，每个产品通过增加插件在工作台上增加功能，这样的好处是共享的工作台可为团队中的所有成员提供相同的底层数据视图，并按照不同角色以可视化形式来展现。这样，就可以为每个开发人员根据其担任的角色来提供适合其需要的工作环境，从而提升开发效率，增强过程管理。

示例场景：在同一个Java项目中，开发人员打开Java视图开发代码；架构师打开建模视图，以UML图形方式展现代码的逻辑结构，并在此基础上复审和修改架构；测试人员打开测试视图，根据代码产生测试用例，启动测试。从这个场景可见，开发团队协作效率是非常高的，因为虽然工具给团队不同角色的人员呈现不同的视图，但是这些视图是基于在同一套底层代码上面，团队的沟通因而也基于相同的基础，沟通更顺畅；另外，团队成员的沟通也无需因为采用不同的工具而不得不进行频繁地导入导出数据工作。

示例场景：开发团队需要定制根据经验积累的新设计模式，在Rational新一代的设计和构建工具中，由于完全基于开放的Eclipse和采用插件的架构，团队可以通过提供的开放的API，方便地产生新的模式，并以插件的形式集成到工作台上，以菜单的方式呈现给设计人员。



图1 IBM Rational Software Development Platform
IBM软件交付平台

架构师设计优秀架构的必备工具

常见问题: 建模工具除了用来画UML图还能做什么?

常见问题: 如何通过Rational的设计和构建工具实现SOA?

Rational Software Architect(RSA)是一个为软件架构师和分析设计人员服务的架构设计工具。RSA目的在于帮助你创造体系结构优秀的应用。越来越多的客户要求建模工具除了能描绘系统的架构,更重要的是还要能验证架构的质量。模型分析: 根据内嵌的UML规则工具自动地对模型进行检查,寻找其中违反规则的模式结构,如循环依赖关系,提高架构师和分析设计人员的工作效率和模型质量。架构复审: 通过对代码的复审,反向了解系统的架构,根据工具提供的“错误架构模式”来了解系统的可靠性和健壮性等。通过UML Profile for Software Services这一插件,您可以使用RSA进行SOA服务的设计,描绘服务的接口、内容以及服务间的相互协作关系,并通过RSA的UML->Java->Web Services的转换,最终实现服务的构建。

示例场景: 架构师通过RSA定义系统的高层次架构模型;在高层次架构的指导下,分析设计人员通过RSA进行子系统、类和接口的设计;分析设计人员将设计递交给架构师,架构师通过RSA的模型分析进行验证。

示例场景: 架构师和分析设计人员通过RSA定义系统的模型;架构师和分析设计人员根据模型生成代码;开发人员细化代码,通过代码复审确认符合代码规则;开发人员将代码交于架构师,架构师通过架构复审确认修改后的代码符合架构规则。



图2 RSA架构及代码复审功能界面

开发人员快速开发的必备工具

常见问题: 开发人员只需要一个基本的代码开发工具就可以了,开发工具的很多功能都似乎没有什么用处?

Rational Application Developer(RAD)给Java和J2EE开发者提供了可视化编辑等手段及一系列的模板和拖拽控件等,务求帮助开发人员用最短的时间完成常规任务,解放生产力,让他们可以投入最重要的业务逻辑开发当中。RAD提供了以下快速开发的功能:

- Web应用程序快速开发和部署: 通过网站设计师工具等工具,不需要写Java码,就能很快地开发一个网站结构和页面。
- 迅速开发复杂的Java和J2EE应用程序: 通过图形化界面的帮助,能迅速的编写访问数据库的代码,快速形成报表;通过UML模型方式可视化代码,使得开发人员可以更直观地在修改模型的基础上修改代码。
- 支持portal应用程序的快速开发和部署: 开发者能使用RAD向导,产生一个遵照J2EE标准的Portlet项目结构,创建完整的Portlet。
- 支持Web Services的快速发现和开发: 通过UDDI资源管理工具, RAD不需要额外的编程,就很容易找到存在的Web service;通过WSDL文件, RAD迅速生成JavaBean或EJB组件的构架。

示例场景: 开发一个动态网页,需要编写常规代码访问后台数据库。想象一下,如果RAD能帮助您直接通过可视化界面的方式访问后台数据库的表数据,您需要做的只是把您要展现的数据以数据源的方式拖拽到网页上,那么您的开发效率是否能提高很多?您是否节约了大量的常规开发时间,使得您可以更专注于业务逻辑的开发?答案无疑是肯定的。

开发人员代码质量保证的必备工具

常见问题: 测试不是都应该交给专业测试人员去做吗?

常见问题: 测试应该只需要在最后阶段进行集成测试就可以了?

开发人员有责任编写功能代码,同时也就有责任对自己编写的代码进行单元测试。执行开发人员的单元测试,就是为了尽早地发现代码中的缺陷,在早期用较低的成本解决,而不是等到最后集成测试的时候,花费更大的成本发现缺陷和修复缺陷。Rational Application Developer(RAD)为单元测试提供了以下的自动化功能:

- 代码复审: RAD提供了Java/J2EE代码复审功能。根据内嵌的200多条复查规则,开发人员可以快速和方便的在代码阶段找出问题,及时地解决缺陷,从而节省时间,提高开发效率。

- 组件测试: RAD可以为组件快速的创建相关测试而不需要写代码; 通过直观的数据表来确定要将什么数据输入到正在测试的方法当中; 测试完成之后, 将提供一个组织好的测试结果列表。
- 运行时分析: 运行时分析提供了对你的应用在运行时进行X光透视。它让你能够看到表面以下的东西并看到系统运行时其内部的情况。

示例场景: 开发人员在开发了几个Java组件程序, 为了确认代码符合企业定义的代码的最佳实践, 开发人员运行RAD的代码复审功能, 找出那些不推荐的编写代码方式, 并根据建议进行修改。在此基础上为了验证代码模块在既定的输入下, 输出为设想中的结果, 开发人员通过RAD提供的快速测试功能, 运行代码并比较预期输出值与实际输出值是否一致。并且了解代码在运行状态下的状况, 包括调用关系图和代码运行时间图等, 及早地发现代码带来的性能问题。

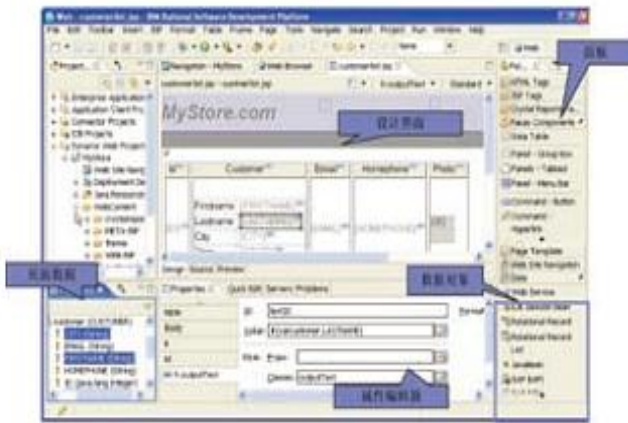


图3 RAD快速开发功能界面

数据库设计师设计高质量数据模型的必备工具

常见问题: 我需要这样的数据建模工具, 通过它的图形界面我可以直观地分析数据源的结构, 并自动地发现分布在不同数据源的同一种信息之间的映射关系。

常见问题: 我需要这样的数据建模工具, 通过它我可以实现从系统模型正向生成或从现有表结构逆向生成数据模型。当数据模型发生变更的时候, 工具会自动地帮我把所有与此变更相关的元素以图形的形式直观地表达出来。

Rational Data Architect(RDA)提供了丰富的数据发现, 开发, 管理功能, 解决您以上的问题, RDA同时支持丰富的数据库类型如DB2, Oracle和SQL Server等。

实现软件开发生命周期的集成

常见问题: 如何实现需求与模型的追踪? 模型与代码的追踪? 当出现需求变更的时候, 您如何通过追踪性准确迅速地找到相应的模型和代码进行修改?

常见问题: 开发模型或代码的时候是否能与配置管理和变更管理工具进行无缝的集成? 以使得在同一个开发环境内, 可以进行模型和代码检入检出工作, 查询和创建变更?

Rational设计和构建产品提供了与需求管理工具RequisitePro/Doors集成功能, 您可以将模型与需求直接关联; 同时, 通过RSA的UML到代码的转换, 产生模型与代码的关联关系。Rational设计和构建产品提供了与配置管理工具ClearCase和变更管理工具ClearQuest及其和新一代配置工具Rational Team Concert的集成功能, 在同一个IDE环境之内, 您可以方便地查询变更请求的情况并进行模型/代码的检入检出工作。

示例场景: 系统分析人员收到需求变更请求, 他/她需要在批准需求变更之前了解此变更对系统架构的影响并估计实现此变更大致需要的工作量。为此, 系统分析人员通过RequisitePro/Doors定位需求, 并通过RequisitePro/Doors与Rational Software Architect的集成与RSA管理的用例模型相关联, 能够定位到用例模型中的用例, 转而继续定位相应的子系统和设计类, 由此评估变更对系统架构的影响, 并估算工作量。

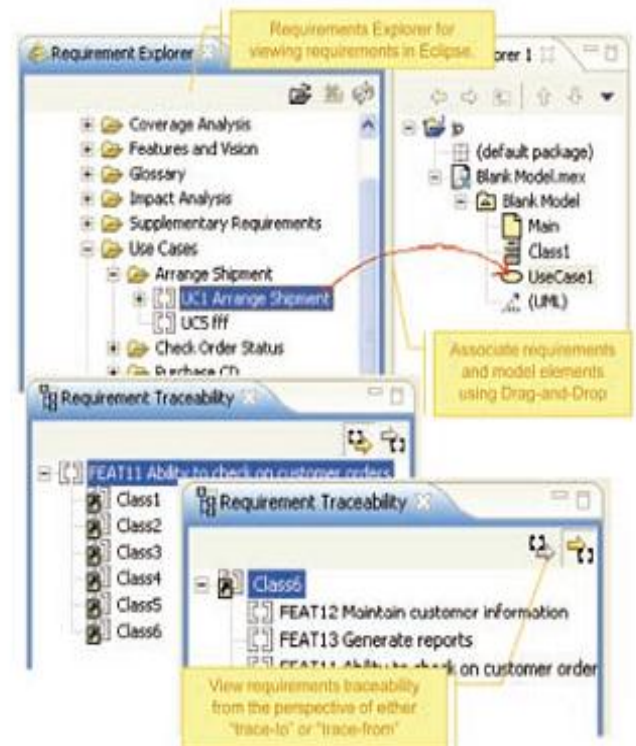


图4 RSA与RequisitePro集成功能界面

设计和构建产品	内容介绍
Java/J2EE/Portal开发 IBM Rational Application Developer	如果您想进行快速的Web/J2EE/Portal/Struts开发和代码质量的控制, 您可以给您的开发人员配备Rational Application Developer <ul style="list-style-type: none"> 包含Rational Application Developer - Community Edition功能 全面支持Java技术和J2EE编程模型 以UML图形方式编辑代码 XML开发, WEB服务寻找和开发 构建数据库驱动的程序 可视地开发Portlets和门户应用程序 自动的装配应用程序 代码复审, 组件测试, 运行时分析 支持与ClearCase/ClearQuest/Rational Team Concert/RequisitePro/RUP/CVS集成 支持Windows和Linux平台
架构设计, 架构质量控制及代码开发 IBM Rational Software Architect Standard Edition	如果您的团队进行Web/J2EE/Portal/XML/Struts开发, 同时, 如果您想建立好的软件架构并对架构进行严格的质量控制, 您可以给您的架构师和高级开发人员配备Rational Software Architect <ul style="list-style-type: none"> 包含Rational Application Developer的功能 包含Rational Software Modeler的功能 架构发现, 分析和控制 创建和应用模型转换(Transformation)和模式(Pattern) 支持Windows和Linux平台
架构设计, 架构质量控制及代码开发 IBM Rational Software Architect For WebSphere Software	架构师和设计师人员通过此工具进行WebSphere Application Server和WebSphere Portal Server上的建模, JEE和SOA开发以及模型与代码的同步
数据结构设计 Rational Data Architect	如果您进行数据结构的开发和设计, 您可以选择Rational Data Architect <ul style="list-style-type: none"> 图形化界面呈现各种数据源的数据结构 提供正向和逆向的方式快速开发数据结构 实现不同数据源之间的数据映射 评估数据结构变更带来的影响 支持与ClearCase/ClearQuest/RequisitePro/RUP/CVS集成 支持Windows和Linux平台
架构设计和代码开发 IBM Rational Rose Developer for Visual Studio IBM Rational Rose Developer for Java	如果您进行Visual Studio或者Java的开发, 您同时希望在一个工具里面进行架构设计, 模型与代码的同步及代码开发, 您可以选择Rational Rose Developer for Visual Studio或者Rational Rose Developer for Java <ul style="list-style-type: none"> 支持UML 1.4标准 Rational Rose Developer for Visual Studio支持UML到ANSI C++, C++, CORBA, Microsoft Visual Basic (MSVB) Microsoft Visual C++ (MSVC++)转换 Rational Rose Developer for Java支持UML到Java, J2EE转换 支持与ClearCase/RequisitePro集成 支持Windows, Unix和Linux平台 注: Rational Rose Developer与新一代Rational Software Architect(RSA)的区别在于RSA完全基于Eclipse, 支持最新的UML2.0标准, 支持Java, J2EE, Portal, Web Services或C++的开发, 具有强大的架构和代码质量控制及运行时分析的功能, 我们推荐用户如果在Windows或Linux环境建模并同时开发Java, J2EE, Portal, Web Services或C++, 直接选择RSA; 但如果用户需要在Unix环境下建模和开发, 或是采用Visual Studio, Dephi等工具进行开发编码, 选择Rational Rose Developer.
架构设计和代码开发 IBM Rational Rose Enterprise	如果您进行Visual Studio和Java的开发, 您同时希望在一个工具里面进行架构设计, 模型与代码的同步及代码开发, 您可以选择Rational Rose Enterprise, Rational Rose Enterprise = Rational Rose Developer for Visual Studio + Rational Rose Developer for Java + Rational Data Modeler <ul style="list-style-type: none"> 包含Rational Rose Developer for Visual Studio功能 包含Rational Rose Developer for Java功能 数据结构建模 支持与ClearCase/RequisitePro集成 支持Windows, Unix和Linux平台 注: Rational Rose Enterprise与新一代Rational Software Architect(RSA)的区别请参考以上Rational Rose Developer与RSA的比较,
嵌入式系统架构设计和开发 IBM Telelogic Rhapsody	如果您进行的是嵌入式系统的开发, 您可以选择建模和代码开发工具IBM Telelogic Rhapsody <ul style="list-style-type: none"> 全面支持UML2, SysML, DoDAF和AutoSAR 用先进的DMCA(动态模型代码自动关联)技术 支持基于宿主机和目标机的仿真 自动代码生成技术, 可以根据模型自动生成C/C++/Java/Ada代码
企业架构和业务流程分析工具 Telelogic System Architect	具体请参见Telelogic System Architect 解决方案