

使用 IBM Rational 软件开发平台
进行 SOA 开发: 实践指南
2005 年 9 月



Rational software

使用 IBM Rational 软件开发平台 进行 SOA 开发: 实践指南

Alan W. Brown

Simon K. Johnston

Grant Larsen

Jim Palistrant

目录

1	目录
2	从业务角度看待 SOA 开发
2	对业务灵活性要求
2	IT 灵活性造就业务灵活性
3	从单体式应用到复合式应用
4	设计和构造的新观点
4	面向服务体系架构 (SOA): 面向变化的蓝图
6	过渡到 SOA
6	面向服务的设计和开发
7	业务驱动的 SOA
8	资产生命周期管理
9	IBM 和 SOA 开发
10	SOA 的业务驱动的开发
11	针对 SOA 的 IBM Rational 软件开发平台
13	IBM 的 SOA 编程模型
14	业务建模和需求
15	面向服务的建模和设计
16	面向服务的构造
18	面向服务的复合和组装
19	过程和组合管理
20	软件质量
22	资产生命周期管理
23	一个 SOA 业务驱动开发的例子
24	IT 主管和项目经理: 调整、执行和控制投资
26	业务分析员: 分析和设计业务流程
27	软件架构师: 设计业务服务
29	开发人员: 构造服务
31	测试人员: 测试服务
31	集成开发人员: 组装和部署服务
32	项目经理和团队: 管理资产生命周期
33	项目经理和团队: 利用最佳实践和指南
33	从 IBM 和 SOA 开始
35	结束语

从业务角度看待 SOA 开发

对业务灵活性的要求

正如过去几年中所发生的, 当今的业务发展领先的企业都在努力地增加收入、降低成本、提高人员和流程的效率。当前的业务和过去的业务的主要区别是速度更快、不断变化。那些能够以敏捷性和灵活性响应变革的企业将能够从那些具有复杂的、僵化的流程和系统的企业中脱颖而出。灵活的业务容易适应任何形式的变化, 包括合并和并购、遵循法规要求、竞争、不断演化的技术, 以及在外包和内购中不断变换的机会。

业务发展领先的企业必须不断地重新考虑、重新定义并采用他们的业务模型、运营方式和企业。那些无法改变或者不愿意改变的企业, 将被迅速地抛在后面直至淘汰掉。

IT 灵活性造就业务灵活性

在随需应变的世界里, 成功依赖于快速响应新挑战和新机遇的能力。一个随需应变的企业能够有效地转换业务模型和流程来适应变化。一个随需应变的企业能将业务与技术很好地协同以最大化灵活性和响应能力, 以建立一个完全整合的、现代化的业务。如果企业能够弥补业务视图中的活动及流程与实现这些活动和流程的技术之间的裂痕, 则它们能够随 IT 方案的进步来进化业务模式。为了弥补业务和 IT 之间的裂痕, 实现业务目标, 必须确保通过目标和需求来推动 IT 开发项目。此外, 拥有了灵活的技术基础设施, IT 就会成为响应能力和适应性的推进器, 而不是一个障碍。

从单体式应用到复合式应用

传统的企业级应用往往尺寸较大，采用单体式解决方案，针对的是某个特定的业务功能。设计技术、工具和流程等均针对开发这种类型的应用而优化。然而，当今的业务系统越来越多地涉及复合式应用。这些 n 层的应用包含了许多整合的功能，使用来自于多个源的信息和逻辑。为了使这些功能协同工作，复合式应用会衍生出多个事务和子事务并且跨各种运行时平台和系统。

例如，如图 1 所示，一个复合式应用可能服务于一个正在与一个复杂系统进行交互的最终用户，这个复杂的系统包括了 Web Services 器、J2EE 应用服务器、集成中间件、以及旧有的系统。这种架构是当今许多企业系统所拥有的典型架构。

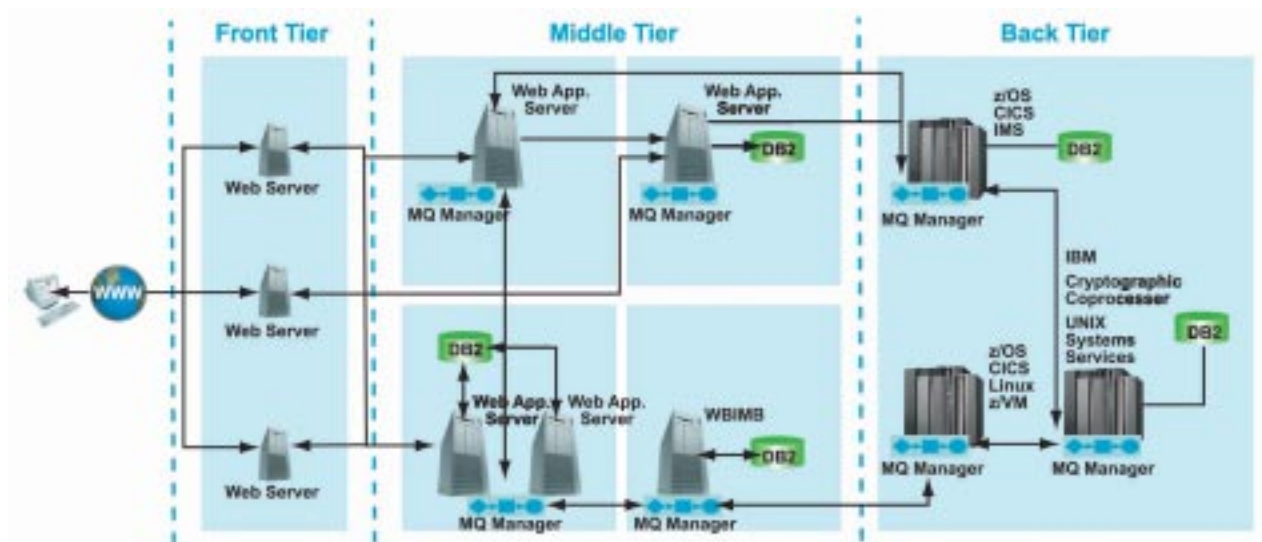


图 1. 一个跨 Web Services 器、J2EE 应用服务器、集成中间件、及旧有系统的复合型应用。

在这种复杂环境中，建立、运行和管理应用程序变得越来越困难。来自世界的分析员的当前数据表明软件开发工具和技术正无法满足这种环境中面临的许多挑战，企业明显需要新方法来开发和管理复合式应用。设计和构造的新视点灵活性、与业务同步和效益的推动力正在改变着企业看待下一代 IT 解决方案的视点。

设计和构造的新视点

灵活性、与业务同步和效益的推动力正在改变着企业看待下一代 IT 解决方案的视点。以下三大推动力体现了这一新视点：

- 将 IT 基础设施的能力视为一系列服务，这些服务可以组合起来满足特定的业务需求。这帮助更好地洞察业务流程如何在 IT 解决方案中实现。
- 将组件组装和分解成作为业务和市场条件要求的复合式应用。
- 重点关注资产管理和重用，以获得更大的业务效益。

面向服务体系架构 (SOA): 面向变化的蓝图

用灵活的 IT 解决方案来响应不断变化的业务需要导致许多企业转向面向服务的体系架构 (SOA)。SOA 是组合了独立的业务功能和流程 (称之为服务) 来实现复杂业务应用和流程的一个 IT 框架。SOA 是一种 IT 方法，它把业务流程视为独立于应用程序及其运行的计算平台的可重用组件或服务。它允许您将解决方案设计为对服务的组装，其中装配的描述是可管理的、解决方案的重要表象，可以对其进行分析、变更和进化。然后，您可以将解决方案视为一组经过编排的服务之间的交互。这种将企业解决方案视为将服务通过明确定义的接口连接起来组成服务联盟的观点正得到广泛的支持。采用 SOA 的最终目标是通过 IT 获得业务灵活性。

已经定义了几种重要的技术和标准来支持SOA方法,特别是当服务在多个机器间分发和通过 Internet 或 Intranet 连接时显得尤为突出。结合了 Web Services 的SOA依赖于内部服务通信协议,例如简单对象访问协议 (SOAP);用 Web Services 定义语言 (WSDL) 表示 Web Services 接口;在可重用资产规范 (Reusable Asset Specification, RAS) 库、目录和统一描述、发现、集成 (Universal Description, Discovery, and Integeration, UDDI) 中注册、搜索、重用服务和相关的资产;在用可扩展标记语言 (XML) 定义的文档中共享信息。

虽然可以创建一个不使用 Web Services 技术的 SOA,并且可以以一种非面向服务的方式使用 Web Services 技术,但当今 Web Services 是实施 SOA 的最常用方式。

当然,SOA 不仅是一套标准和服务的描述,同样面向对象架构也不仅仅是一组类的层次结构。SOA 的灵活性体现在它将业务流程和底层 IT 基础设施中的元素看作安全的、标准化的组件 (Web Services 是其中对典型的存在形式),通过对这些组件进行重用和组合,即可应对不断变化的业务目标和业务优先级。SOA 使企业能够应对关键挑战,快速而有效地挖掘重要的机会。它还:

- 使得将遗留系统链接到新应用更加容易。每一波技术潮流——从大型机到客户端/服务器数据库、企业资源规划 (ERP)、Web 应用、Java 应用和当今的服务——已经贡献了许多应用,大多数企业可能在未来使用这些应用。SOA 简化了新应用与一个或多个“遗留”应用程序的集成。

- 帮助改进内部业务流程。分析、改进、自动化或创建新业务流程对于节约成本、提升客户满意度和每项业务的主动性是非常重要的。用 SOA 组装或分解解决方案的能力流线化了这些工作。
- 增强共享业务流程。SOA 基于开放的标准，具有平台无关性，这使企业能够自动化通用的 B2B 流程，并且使提高供应链效率更加容易。

过渡到 SOA

过渡到 SOA 是一个频繁需要业务经理和 IT 经理切换视点的旅程。特别是，他们不仅需要把重点放在一个新的、面向服务的方法上来开发应用，而且要把重点放在业务驱动的开发和资产的生命周期管理上。

面向服务的设计和开发

面向 SOA 创建解决方案的企业必须重新考虑他们正在建立的系统的种类、重新评估他们组织中的技能、重新定义团队成员协作的方式。更为重要的是，采用面向服务的方法来开发解决方案必须重新考察它的影响，这些影响包括您如何设计解决方案、组装不同的服务意味着什么、您如何管理和改进部署的面向服务的应用程序。IBM® 将这种更广泛的面向服务的环境称为面向服务的设计和开发。

面向服务的设计和开发涉及一系列广泛的能力、技术、工具和技能的集合，包括：

- 管理服务的生命周期——包括发现、应用、改进和维持服务。
- 建立一个平台和编程模型，这包括在特定的运行平台中连接、部署和管理服务。
- 采用能使团队在解决不断变化的业务需求方面有效地创建和组装服务的实践和工具。这包括提炼现有的应用程序来发现潜在的服务，包装现有的功能来使这些功能能够作为服务被访问，创建新服务，通过连接由其接口公开的行为来将服务“连”在一起。这些功能的基础是具有可用的清晰指导和以可重复且可预计的方式构造面向服务解决方案的最佳实践。

业务驱动的 SOA

在 IBM 称为 Business-Driven Development for SOA 的解决方案中，业务目标和需求推动下游设计、开发和测试，从而将业务流程转换为对业务进行了自动化和整合的复合式应用。需要跟踪整个生命周期中的需求，从业务目标到软件设计与编码资产，再到复合式应用。

应该设计整合的解决方案，确保它是灵活的，能够随着企业需求变更而适应；应该最大限度地提高资产重用、减少冗余；最终，从头开始高质量地进行构建。

业务驱动的开发缩短了业务和 IT 之间的距离，确保基于业务需求建立解决方案。同时，交付的解决方案更加灵活，使业务能够迅速理解不断变化的业务需求所造成的影响，并进行适当的响应。

资产生命周期管理

当企业将服务作为系统设计中的关键资产时，重用这些服务的价值就变得更加明显了。因此，资产管理和宏观管理的技术和方法、捕获组合资产模式的可重复方法变得越来越重要。在一种基于资产的开发方法中，这些资产对于企业具有重要的价值，必须仔细地管理。在这种方法中，跨项目和企业一致地管理资产的团队基础设施起着至关重要的作用。

像任何资产一样，服务的生命周期包括多个阶段：识别和发现服务；获取并创建服务；认证和发布服务；重用和度量服务；直至最终撤销服务。一组称为 Asset Production, Asset Identification, Asset Management, and Asset Consumption 的工作流 (参见图 2) 描述了这一生命周期。

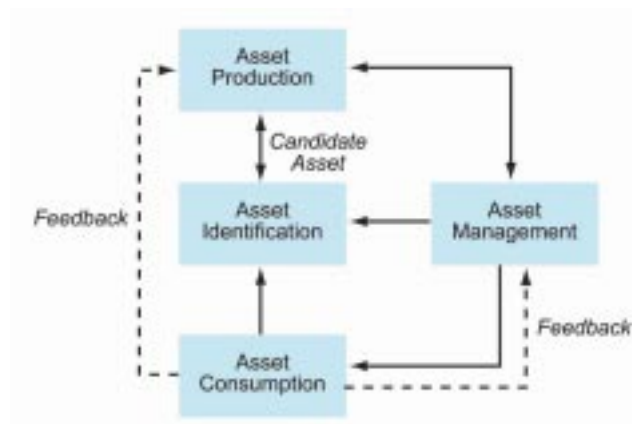


图 2. 服务生命周期管理工作流

可以在生命周期的各个点使用或重用服务，包括开发期间和运行时。为了在开发过程中有效地重用服务，必须将服务与其保持一致的工件和描述服务的元数据共同打包。可重用资产规范 (Reusable Asset Specification, RAS) 定义了用于打包服务的元数据格式，RAS 是一个 Object Management Group (OMG) 标准。RAS 以 XML 语言表述，每个资产具有一个包含元数据的 RAS XML 清单。

这些服务元数据包括分类信息和版本信息，以及描述服务工件和工件之间相互关系的信息。对于开发人员，RAS 元数据描述了如何适当地使用服务和它所使用的上下文。

IBM 和 SOA 开发

SOA 使企业能够提高 IT 灵活性，更好地让 IT 与业务目标保持一致。当然，构建 SOA 解决方案所有工作的一部分。IBM 为 SOA 整个生命周期的支持提供了 IBM SOA Foundation (参见图 3)，包括建模、组装、部署和管理 SOA 解决方案。IBM 还提供培训和咨询服务，以及多年的 SOA 专家经验来帮助客户构建基于 SOA 的解决方案。

IBM SOA Foundation

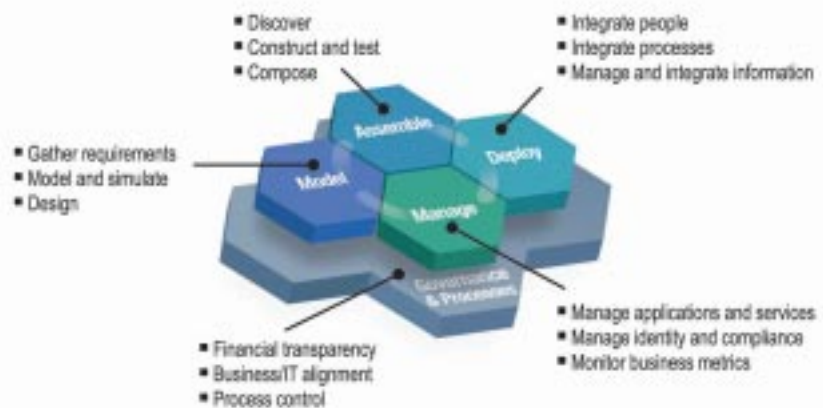


图3: IBM SOA Foundation 是一个模块化的、集成的、开放的软件、最佳实践和模式的集合，它支持整个 SOA 生命周期。

IBM SOA Foundation 重点放在根据业务优先级对 IT 进行调整, 使业务具有灵活性: 可以建模业务流程; 设计和构造服务, 并把它们组装到复合式应用, 然后可以部署、管理它, 并在需要满足变化的业务需求时进行适应。

本文的后续部分将重点关注 IBM SOA Foundation 的构建方面, 您可以建模、构造、测试和组装解决方案。

SOA 的业务驱动的开发

使用业务需求来推动所有下游的开发活动时, 缩小了 IT 和业务之间的距离。在面向服务的设计和开发环境中, 结果是优秀的、灵活的解决方案, 您能够迅速地适应不断变化的业务需求和优先级。SOA 的业务驱动的开发 (图 4) 提出了业务经理正面临的两个关键挑战——提高响应能力和让 IT 适应业务目标。

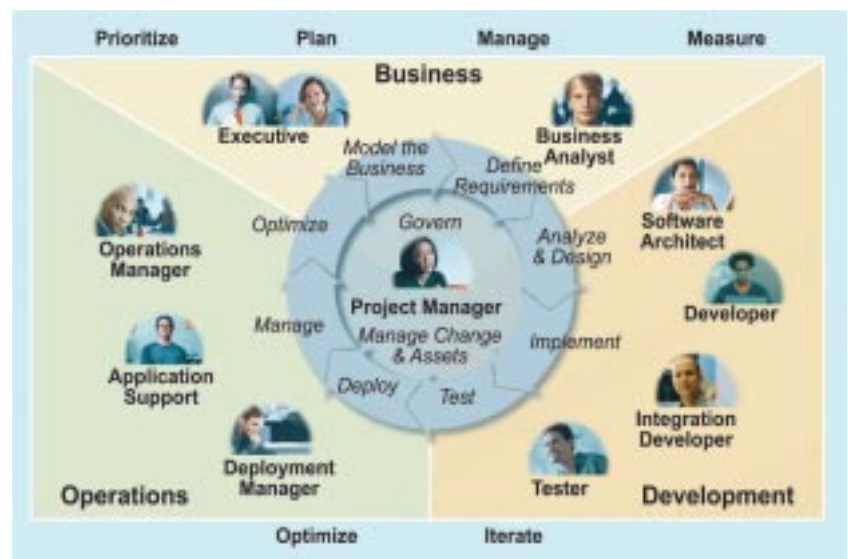


图 4: SOA 的业务驱动的开发将业务流程转换为对业务进行自动化和整合的复合式应用。

针对 SOA 的 IBM Rational 软件开发平台

IBM Rational® 软件开发平台为开发团队提供了一个完整的、开放的、模块化的和经过验证的环境，可用于针对灵活的 SOA 解决方案的业务驱动的开发。它包括：

- 基于标准的、使业务流程建模的许多方面实现了自动化的新型编程模型，以及服务设计、构造和组合，这些因素都削减了项目时间和成本。
- 行业领先的过程和产品组合管理以及软件配置管理解决方案，这些解决方案引导从业者通过 SOA 开发生命周期的每个阶段，最大限度提高资产可重用性、促进项目成功和确保可预测性。
- 基于角色的工具为业务和技术用户提供了各自需要的准确的功能集合，加上 Eclipse 支持的紧密集成，通过促进协作缩小了业务和 IT 之间的差距，确保业务需求推动下游开发，并提供了从开始到部署过程中对需求的跟踪能力。

IBM Rational 软件开发平台是针对 SOA 业务驱动的开发的强大基础。它还支持以代码为中心、可视化编程、模型驱动等开发风格。

除了为每个开发生命周期阶段提供广泛的工具外，为了支持解决方案的面向服务视图，面向服务的 IBM Rational 软件开发平台还通过工具深层次集成和运行时功能为业务的各个方面提供真实的价值。

IBM Rational 软件开发平台中工具的紧密集成部分基于 Eclipse 平台。Eclipse 技术使得在单一且一致的软件开发平台中实现集成性、灵活性和可扩展性。

它作为更广泛的生态系统的基础，使 IBM Rational 软件开发平台可为众多用户使用。

如图 5 所示，IBM Rational 软件开发平台为建立 SOA 应用程序提供了特定于角色的工具，包括需求和分析、设计、构造、软件质量、过程和项目组合管理、资产生命周期管理。下面概述了每个功能，包括它们如何适用于 SOA 方法，以及支持它们的 IBM Rational 软件开发平台工具。这个概述将为更高层次的讲解这些工具如何在一个示例 SOA 项目的开发周期中协同工作打下基础。



图5 针对 SOA 的 IBM Rational 软件开发平台

IBM 的 SOA 编程模型

面向服务解决方案的开发要求团队来定义、管理并不断进化一组不同的元素。这些元素描述了正在操作的永久性数据如何表示将要执行的业务算法, 支持与系统用户进行有意义的交互。总体来说, 这些元素的集合通常称为编程模型。

IBM Rational 软件开发平台的一个关键方面是使用受 SOA 强烈影响的编程模型, SOA 在 Web Sphere® 平台、IBM 中间件的堆栈 (DB2®, Tivoli® 和 Lotus®) 中得到实施, 特别是在 IBM Rational 软件开发平台中。

IBM SOA 编程模型将开发和部署活动分解为能够发生在不同时间的阶段, 具有不同技能的个人都能够执行。这改变了我们对于应用的关注点, 使您能够重新计划软件组件。

IBM SOA 编程模型的关键元素包括:

- Service Data Objects (SDOs) (现在处于 Java Community Process 的标准化阶段中); 它们为各种资源提供一个简化的数据访问编程模型, 并且补充了核心的 Web Services 标准, 如 XML、WSDL 和 SOAP
- Business Process Execution Language for Web Services (BPEL*), 它是一个支持工作流和业务流程集成的服务编排和组件脚本标准
- JavaServer Faces (JSF), 它是为非专业 J2EE 开发人员提供的加速 Web 应用程序开发的一个 Java 框架

*BPEL 现在正在被标准化为 WS-BPEL

- 使用外部策略和规则来定制应用程序, 这些策略和规则包括一系列在策略定义和执行的开发中新出现的标准, 包括 Web Services Policy and OMG Business Semantics of Business Rules (BSBR)

一个描述 SOA 解决方案的总体服务连接和组装方面的简化模型对于创建 SOA 解决方案是非常关键的。IBM 也引入了服务组件架构 (Service Component Architecture, SCA), 这是一个简化包含各种服务实现技术的抽象化服务。SCA 补充了 SDO、BPEL、业务状态机和业务规则支持, 简化和加速服务设计、开发、组装和部署。

业务建模和需求

有效的业务驱动的开发从分析业务的实际需要、需求和现有流程开始。然后, 业务分析员使用该分析来推动需求和所有后续开发阶段。

WebSphere Business Modeler 用于快速而高效地对复杂业务流程进行建模、模拟和分析。分析员使用 WebSphere Business Modeler 来建模 “as-is” 和 “to-be” 业务流程、分配资源和执行 “what-if” 模拟, 以最佳化和评估业务受益。之后, 他们可以将这些模型转换为 Unified Modeling Language (UML) 和 Business Process Execution Language (BPEL), 以启动设计和集成活动。

一旦分析员通过选择和优化最佳的 “to-be” 流程确定了解决方案, 他们就定义了业务需求, 并在 **IBM Rational RequisitePro** 中把它们精化成软件需求和用例。需求和需求管理对于任何成熟的软件开发过程都是非常重要的, 业务建模帮助确保软件需求和用例反映真实的业务需求。编写较好的用例为

架构、分析、用户界面 (UI) 设计和测试提供了基础。Rational RequisitePro 中的需求能够链接到过程模型，以及测试用例和测试计划，提供完整的可跟踪性，并确保业务需求推动下游开发。结果是一个满足业务目标和需求、提供实际业务价值的系统。在 IBM Rational 软件开发平台中，需求与缺陷跟踪、设计、开发和测试工具集成在一起来启动活动、最小化重复工作，并为每个团队成员提供一个直接窗口输入用户需求。

面向服务的建模和设计

一种设计面向服务应用程序的有效方法需要您在建立服务之前为所有主要的服务组件建模并定义具有良好文档记录的接口。

IBM Rational Software Architect 是一个用于创建面向服务应用程序的设计和构造工具。通过利用 UML 模型驱动开发，将软件应用程序架构的所有方面统一起来，Rational Software Architect 使软件架构师能够设计灵活的服务架构，在从分析和设计到实施的过程中自动为 SOA 应用设计模式。

IBM Web Sphere Business Modeler 和 IBM Rational Software Architect 的集成使团队能够将业务模型转换为自动化业务活动的软件模型。集成自动将业务表示法映射为软件表示法 (UML)，减少设计人员的转换负担，确保软件模型不会不小心对业务流程造成更改。

为了在面向服务的世界中辅助解决方案的设计，通过用特定领域的表示法和相关语言来定义、组装、管理服务，可以定制 **IBM Rational Software Architect**。通用机制可用于创建使用内置产品扩展机制的自定义表示法。使用这些机制创建的特定于域的语言的例子包括：

- **UML Profile for Software Services** 通过提供描述服务的通用语言加速业务流程转换为 **Web Services** 的速度，帮助软件架构师建模、映射和划分服务。
- **UML Profile for Business Modeling** 提供了一种用 **UML** 捕获业务模型的语言，**IBM Rational Unified Process** 中的业务建模方法支持该 **UML Profile**。

面向服务的构造

Java、J2EE 和 Web 开发人员经常执行创建支持 SOA 的核心元素 (通常是 **Web Services**) 的任务。手工创建 **Web Services**，或者甚至是开发使用现有 **Web Services** 的系统，需要大量单调乏味且易于出错的工作。

IBM Rational Application Developer 是一个全面集成的面向服务的开发环境，它将许多在 **Web Services** 创建和使用过程中通常要执行的任务自动化。开发人员能够致力于编写业务逻辑代码，而依赖 **Rational Application Developer** 实现从 **WSDL** 文件和代码生成到测试客户端生成和 **WS-I** 一致性验证。开发人员使用 **Rational Application Developer** 创建、生成、使用、测试、部署和发布 **Web Services** ——要么从头开始，要么通过启用现有应用程序的 **Web Service Interoperability (WS-I)** 一致性。

Rational Application Developer 极大地简化了基于需求规格说明和 UML 模型创建 Web Services 的过程。通过使用 Rational Application Developer 自动生成包含描述 Web Services 的 XML WSDL 文件、JavaBeans 或 Enterprise JavaBeans 框架文件, 开发人员节省了时间, 减少了错误。Rational Application Developer 还包含了一个 XML schema definition (XSD) 编辑器用于指定消息格式。

当从现有的 Java 类或 EJB 自下而上进行开发时, Rational Application Developer 也能够自动化许多将组件转换为 Web Services 的过程。一个易用的向导能够为 Java 类中的方法生成 WSDL 接口, SOAP 部署描述符和 Web 页面可以作为测试客户端, 用于测试与 Web Services 的交互。

Rational Application Developer 还具有 J2EE Connector Architecture 工具, 该工具能让开发人员容易地将现有的 Enterprise Information Systems (EIS), 如 CICS 或 IMS, 集成到 SOA 解决方案中。开发人员能使用这些工具快速地从现有 EIS 事务创建 Web Services。

Rational Application Developer 中对 Enterprise Generation Language (EGL) 的支持使面向业务的开发人员和过程开发人员可以使用过程编程结构, 无需知道 Java 即可开发、测试和调试数据驱动的 Web 应用程序、Web Services 和业务逻辑。EGL 提供了全面的 Web Services 支持, 可处理开发和使用 Web Services 中的大部分艰苦工作。使用现有的 Web Services 或启用现有资产只需要极少的代码, 这将使面向业务的开发人员只需花费较短的学习时间, 提高了生产力。

为了使开发人员在开发周期早期就注重质量, Rational Application Developer 包括:

- 自动的代码检查工具用于验证代码是否最优
- 组件测试工具基于 WSDL 文件分析自动创建测试代码、测试用例并输入数据
- 用于检测和诊断瓶颈、死锁和紊乱情况的性能分析工具信息

Rational Application Developer 还能自动化发现和使用现有 Web Services 的过程。开发人员使用内置的 UDDI Explorer 工具来浏览 UDDI 注册中心、发现 Web Services, 并将它们导入到项目中。

对于 Web 设计人员, Rational Application Developer 将自动生成 Web 页面与 Web Services 进行交互所需的所有内容。它使用 JavaServer Faces UI 控件为 Web Services 输入和输出生成 JSP 代码, 并生成调用 Web Services 所需的 Java 代码。

面向服务的复合和组装

集成开发人员通过将业务流程 (在 Web Sphere Business Modeler 中创建) 精化为一套可执行的、精心编排的服务来创建和部署复合式应用。集成开发人员使用 **IBM WebSphere Integration Developer** 来组装部署到 IBM Web Sphere Process Server 的复合式应用。使用可视的 BPEL 编辑器, 集成开发人员能够查看业务分析员设计的业务流程。然后他们使用 WebSphere Integration Developer 来精心编排服务, 将它们连接成一个复合式应用, 并可使用内置的测试环境测试它们 (可选), 然后部署到一个运行时环境。

Web Sphere Integration Developer 为集成开发人员提供了可视化软件开发工具, 他们需要该工具指定、测试和部署可执行的业务流程, 这些流程将 Web Services、企业应用程序、人工任务和其他服务组件集成到有效的基于 SOA 的业务解决方案中。

过程和组合管理

除了用于面向服务开发的一套完整工具之外, IBM Rational 软件开发平台还包括了实用的过程指南。**IBM Rational Unified Process**, 或 **RUP®**, 是一个可配置的软件开发过程平台, 提供经过验证的最佳实践和可配置的架构。

由于 RUP 建立在软件工程的最佳实践基础上, 它提供一个可配置的过程框架, 并且在支持企业项目时具有可伸缩性, 因此它很好地适应了 SOA 开发项目的需求。

团队成员可以应用 RUP 来支持完整 SOA 复合式应用和解决方案中每个单独服务的开发。RUP 还为桥接业务和 IT 之间的间隙提供了一套系统的方法。IBM 还提供 RUP 插件, 它是可下载的资产, 包含了各种为特定工具、技术或领域定制的过程组件, 还包括针对 SOA 开发量身定做的插件。

- RUP Plug-In for SOA 将对面向服务架构和面向服务解决方案的支持和特定于 SOA 的概念、准则、活动、工件和工具指导集成到 RUP 框架中。
- RUP Plug-In for WebSphere Business Modeler 更新了 RUP 中利用 WebSphere Business Integration 解决方案的业务建模规则, 为基于 IBM Rational 软件开发平台的业务建模提供统一的方法。

从 SOA 角度来看, 应用程序是动态的实体, 即满足一组特殊业务需求的服务组合。采用这种观点的一个效果是需要企业明确地设计、实施和管理企业的服务集合。主管人员和项目经理使用 **IBM Rational Portfolio Manager** 来获得对 SOA 服务组合的业务效益、成本和风险的洞察力。使用 **Rational Portfolio Manager**, 他们能够划分提议的、现有的和正在构造的服务的优先级, 跟踪服务级的财务状况, 管理 SOA 项目团队的依赖关系; 预测服务创建的需求, 更好地理解 SOA 创建、运作和维护的成本。

软件质量

除了 Rational Application Developer 和 Web Sphere Integration Developer 中的开发人员测试功能, IBM Rational 软件开发平台还包括帮助测试人员连续评估 Web Services 和最终复合式应用质量的工具。他们可以将测试计划和单个测试案例链接到 Rational RequisitePro 中的项目需求, 确保完整的基于需求的测试覆盖率。

SOA 解决方案的有效质量保证依赖于功能和性能测试的管理和执行。

目前有多种商业解决方案能够自动编写测试脚本。这些解决方案的实用性依赖于该工具支持的用户界面技术的覆盖范围。业务流程可能涉及几种用户界面, 从“绿色屏幕”应用程序和不同类型的 Web 应用程序 (包括 ASP.NET, JSP, JSF 和 portlets) 到本机应用程序, 比如邮件客户端和普及设备。测试端到端业务流程通常需要自动引擎来驱动所有用户界面。

IBM Rational Functional Tester 为测试人员提供广泛的技术和开放的测试环境。特别是，它支持使用非专有语言编写脚本: **Java**和**VB.NET**。另外，它们能够轻易地按照不同用户界面组合记录测试脚本，这些脚本可以用不同技术创建。这使端到端业务流程的编排验证实际上是无缝的。它们能够从命令行自动触发测试执行，使得将测试集成为端到端生成-部署-测试过程的一部分更加容易。

尽管功能测试对于任何测试工作来说都非常重要，仍需要考虑功能测试之外的几种测试方式。在 **SOA** 应用程序上下文中，有效的性能和安全测试是至关重要的。过多的通信服务和大消息负载能够严重影响 **SOA** 应用程序的性能。因此，需要尽量早在设计期间就处理性能确认问题，**QA** 团队必须着重考虑这个问题。

IBM Rational Performance Tester 是一个测试创建、执行和分析工具，在多种用户负载的情况下验证应用程序的可扩展性和可靠性。**Rational Performance Tester**使团队能够在应用程序部署前正确发现系统瓶颈。因为测试通过记录 **Web** 浏览器中的用户活动来创建，团队需要很少，甚至不需要任何编程知识就能够理解和修改这些测试。数据池容量确保每个模拟用户具有唯一的数据。使用直观的图形测试调度程序，团队能够组织测试，以准确地模拟不同类型的用户和测试应用程序支持的用户活动。

在测试执行期间，当模拟期望的并发用户数量时，**Rational Performance Tester**生成报告来高亮显示性能表现较差的 **Web** 页面、**URL** 和事务。团队能够公开拿怕是最复杂系统的性能问题，增加在系统正式使用前识别和修复问题的机会。

资产生命周期管理

企业把注意力放在资产管理和重用上，考虑它们的下一代解决方案平台。它们正在把重点放在创建、获取、应用和管理整个生命周期的资产上。出于实际考虑，这种关注通常伴随着对可重用资产的强大的宏观管理实践，这些可用资产与灵活的资产生命周期管理系统相关。

随着企业创建 SOA 来利用现有资产，加速和管理变更的能力变得越来越重要。团队使用 IBM Rational ClearCase® 和 IBM Rational ClearQuest® 进行软件配置管理，从而能够实现软件生命周期的自动化，建立一个跨分布式环境的、一致的过程。

IBM Rational 软件配置管理产品帮助团队提高项目协作和发布版本的协调，提高开发响应能力和敏捷性，提高运营效率。

另外，通过对 RAS 的直接支持，可以在 IBM Rational 软件开发平台中打包和图 6. Rational ClearCase 和 Rational ClearQuest 支持 RAS 和资产生命周期管理。使用可重用的 SOA 相关开发资产 (参见图 6)

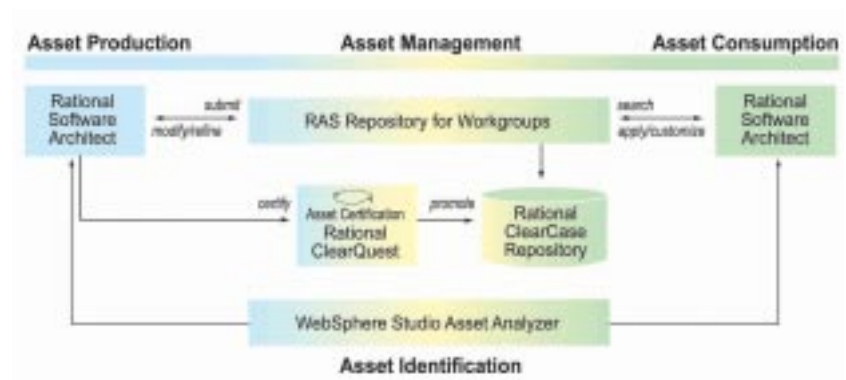


图6. Rational ClearCase和Rational ClearQuest支持RAS和资产生命周期管理。

IBM Rational 软件开发平台包括了其他几个工具，使团队能够更有效地管理资产工作流。WebSphere Studio Asset Analyzer 提供了对 Asset Identification 工作流的支持。Rational Software Architect 为 Asset Production 提供了一个 RAS 客户端。该工具还包括一个称为 Asset Explorer 的 RAS 客户端，用于促进 Asset Consumption。开发人员和架构师使用 Asset Explorer 连接一个或多个 RAS 库，搜索检查每个 RAS 资产中的 RAS XML 清单。

可以通过几种方式来度量资产和服务，包括定量和定性分析。团队可以使用 RAS Repository for Workgroup 来跟踪某些指标，例如用户对一个资产搜索了多少次，他们多长时间浏览其文档，他们导入和重用其多少次。开发人员还可以评估资产或服务的等级，并提供文本形式的反馈。这种度量对于资产库的健康状况和活跃程度是至关重要的。

一个 SOA 业务驱动开发的例子

理解 IBM Rational 软件开发平台在 SOA 业务驱动的开发中的价值的一种有效方法是探索这些工具如何在 SOA 开发周期中帮助各种角色工作协同工作。下面的场景描绘了这样一个工作。

本文保留了实际业务需求的细节。在这个场景中，一家公司可能正在自动化汽车租赁预约过程的某个方面，比如一个工作订单流程，一个供应链或者任何其他业务流程，不管是新的还是现有的，只要是企业需要执行的就可以（参见图 7）。

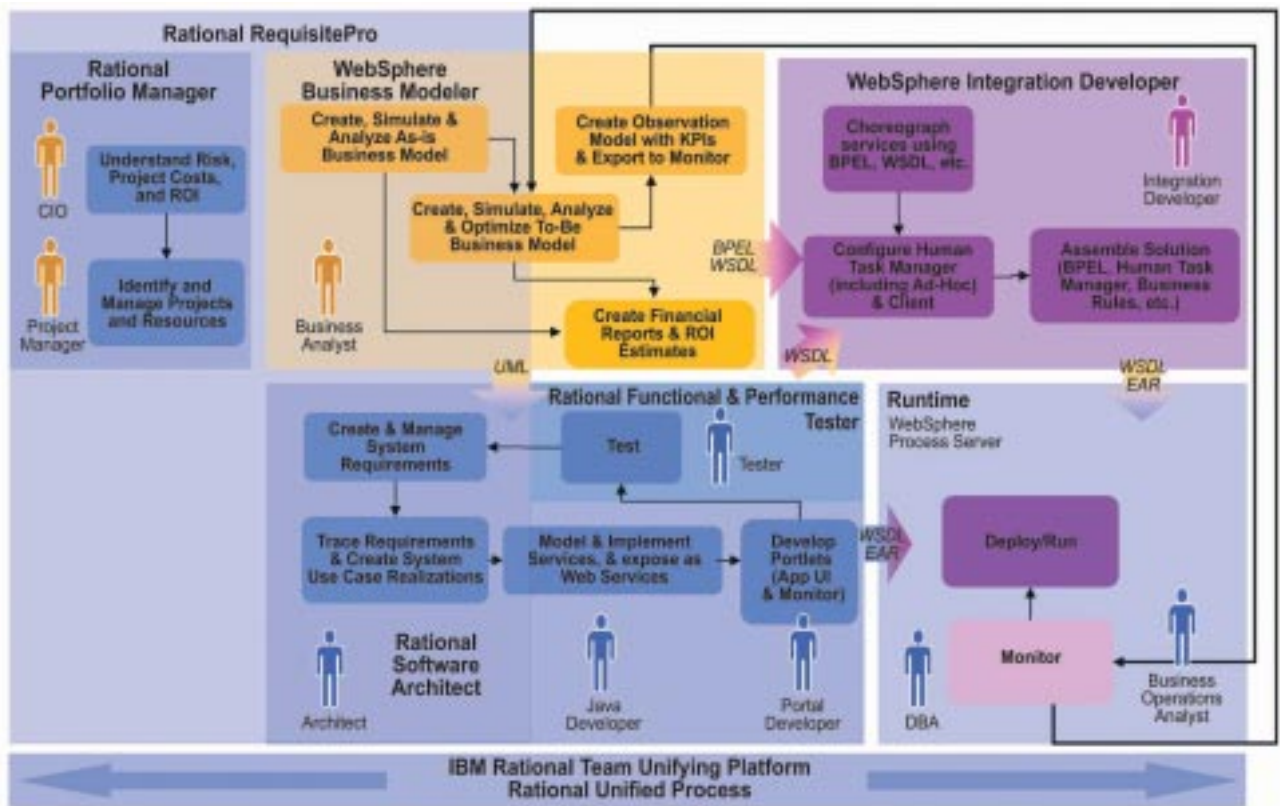


图7. 一个使用 Rational 软件开发平台构建过程集成方案的建议场景

IT 主管或项目经理: 调整、执行和控制投资

宏观管理是 SOA 项目成功的基础。IT 主管和项目经理使用 IBM Rational Portfolio Manager 来评估企业的 SOA 服务组合的业务受益、成本和风险。

通过 Rational Portfolio Manager, 经理可以根据业务优先级、风险和回报情况划分提议的、现有的和正在构建的项目的优先级, 如图 8 所示。另外, 该产品允许他们跟踪服务级的财务状况, 获得对于 SOA 开发的洞察力, 管理 SOA 项目团队的依赖关系, 预测对服务创建和更新的需求。

一旦 SOA 项目开始执行, Rational Portfolio Manager 提供一个完整的项目管理、协作和工作管理环境。它的中央存储库使得许多工作都变得很容易, 如项目经理报告项目状态, 项目内外团队的协作情况, 决策者对 SOA 项目和程序的进度和健康状况的了解, 等等。

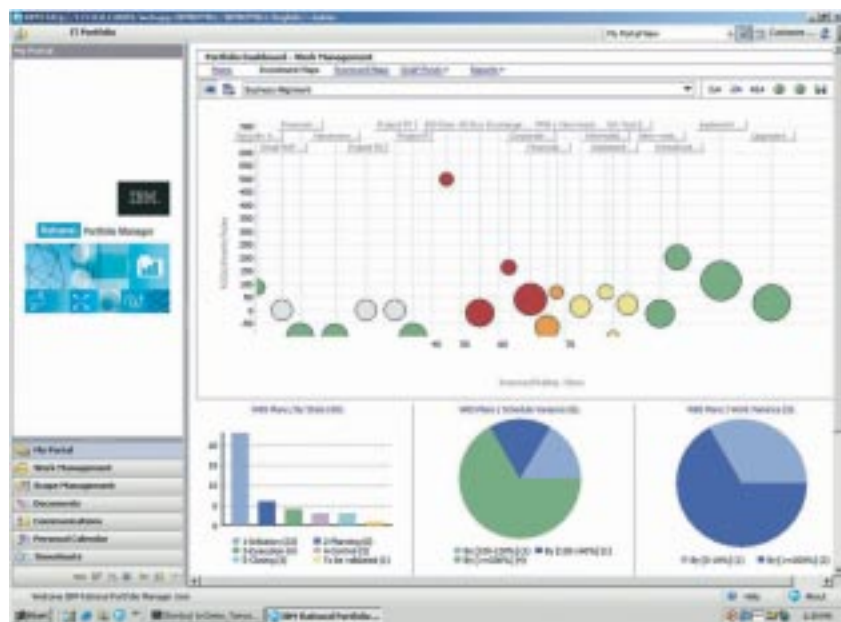


图 8. Rational Portfolio Manager 能够帮助在高层次上评估相关的 ROI、成本和提议的业务调整。

业务分析员: 分析和设计业务流程

业务分析员通过检查推动过程改进请求的业务目标来开始工作。使用 IBM WebSphere Business Modeler, 业务分析员用直观的、易于使用的表示法完成“as-is”过程模型。这使您能够在一个高层次上理解当前自动和手动步骤的系统。为了潜在的投资回报, 企业在任何变更提交到业务流程之前, 应设计、模拟和分析它。这种分析的结果是“to-be”过程模型的开发, 这是 IT 解决方案预期自动化的。

接下来, 为了后续能够被集成开发人员使用, 业务分析员将业务建模项目和工件导出为 BPEL, 也可以是 Web Services Definition Language 和 XML Schema 或 XSD 工件 (参见图 9)。

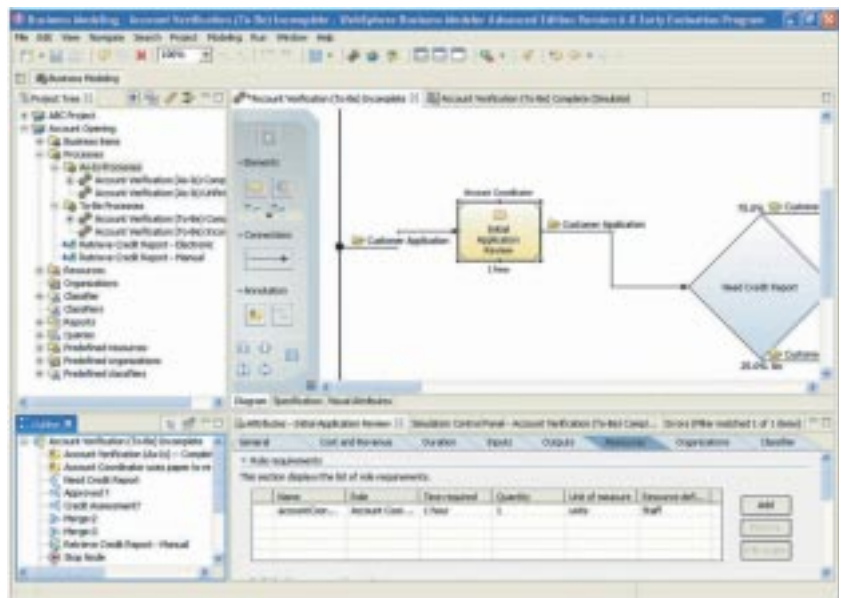


图 9. 业务分析员使用 WebSphere Business Modeler 开发业务流程模型。

与 IT 经理、业务经理、架构师和其他涉众协同工作，业务分析员使用 IBM Rational RequisitePro 来识别那些推动开发生命周期的解决方案需求。

业务分析员和架构师定义业务需求，并使用 Rational RequisitePro 来创造和共享需求，将业务需求精化成软件需求和用例；跟踪需求并执行影响分析；在需求变更时接收及早的通知。

在整个开发过程，Rational RequisitePro 提供给业务分析员和整个团队对集中位置的项目需求的快速、直观访问，使得消息沟通和跟踪更加容易。

软件架构师: 设计业务服务

使用 IBM Rational Software Architect，软件架构师能够导入 Web Sphere Business Modeler 项目，应用模型驱动开发方法，自动将过程模型转换为 UML 模型。然后软件架构师使用 Rational Software Architect 来创建详细的 UML 图，这些图用于识别将要被实施的服务元素。使用 Rational Software Architect 中的 Asset Explorer，架构师还可以搜索和重用整个企业中的现有 RAS 资产，包括服务。RAS 资产定义了一套服务上的丰富数据，包括其 UDDI 注册中心的位置。

架构师能够用对原有业务模型的跟踪能力开发一个设计模型。另外，他在 Rational Software Architect 中开发用例模型，并将用例模型链接到 Rational RequisitePro 中的特定需求；特别是，架构师可能需要识别与从 WebSphere Business Modeler 中导入过程模型相关的非功能需求。

到任何新服务的接口被建模, 显示每个服务期望的输入和输出。(参见图10。)

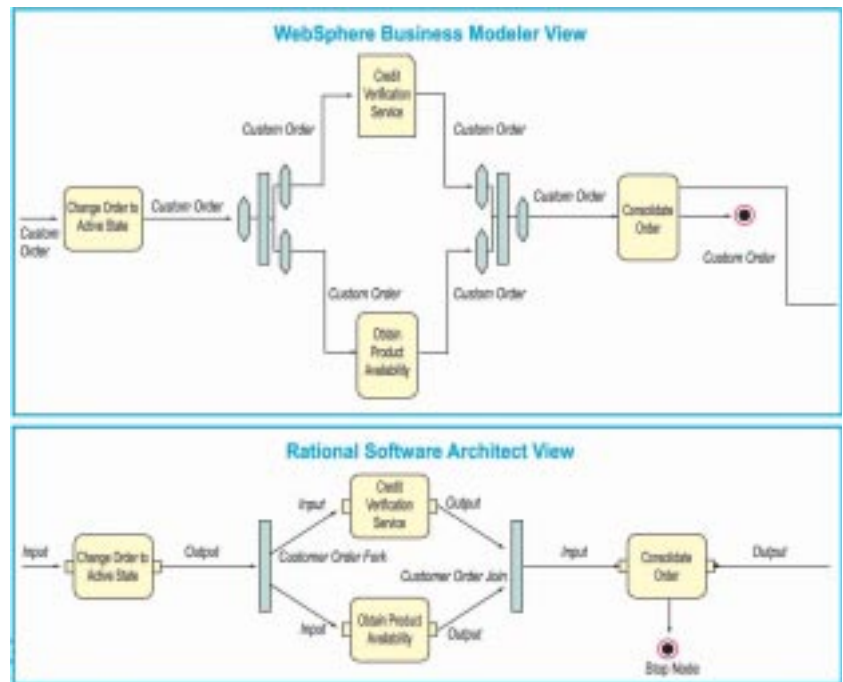


图 10. 业务流程在使用业务相关的表示法在 *WebSphere Business Modeler* 中表示, 以及使用标准 UML 表示法在 *Rational Software Architect* 中表示。

要想开始开发, 架构师在 *Rational Software Architect* 中执行 UML 到代码的转换, 这将生成服务和域模型的骨架, 如图 11 所示。这种转换基于实施了架构师所提供接口的 UML 设计模型。在这种环境中, 接口代表了设计接口的架构师与实接口的开发人员之间的契约。

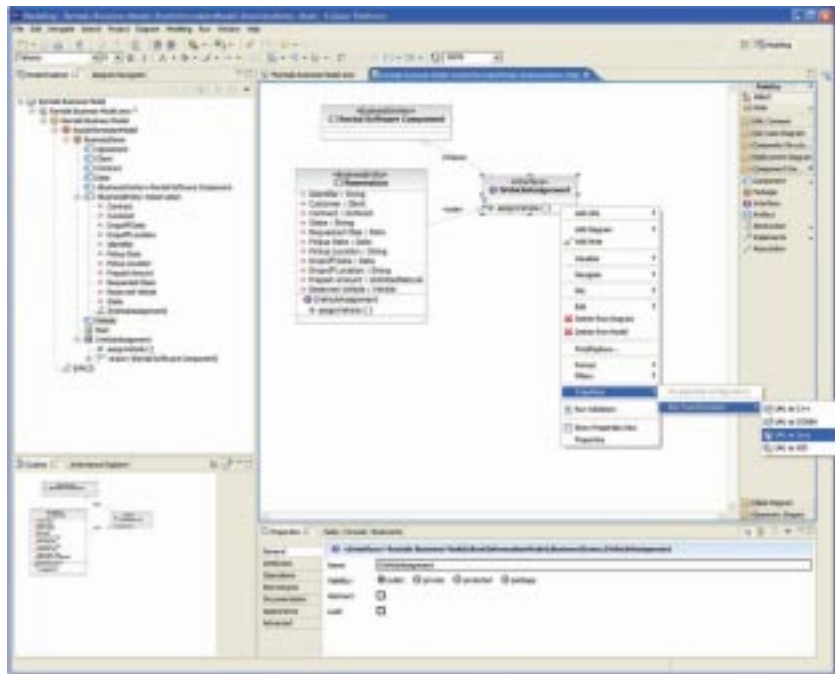


图 11.使用 Rational Software Architect 将 UML 模型转换为 Java 代码。

开发人员: 构造服务

接下来, Java 开发人员使用 IBM Rational Application Developer 实现架构师的规格说明, 并构造服务。

由于许多开发任务都可以自动化, 开发人员可以致力于实现在 Rational Application Developer 中为满足业务功能需求的合适的业务逻辑。然后, 开发人员能够在 Web Sphere Test Environment 中部署和测试代码, Web Sphere Test Environment 包含在 Rational Application Developer 中。

作为一种替代方式, 开发人员可以重新规划现有的资产, 如图 12 所示, 使用 Rational Application Developer 从现有的 Java 类、EJB 或 EIS 资产自动创建 Web Services。在解决方案开发中, Rational Application Developer 和 Rational Software Architect 都能利用工具中包含的 UDDI 浏览器来查找和重用现有的服务。

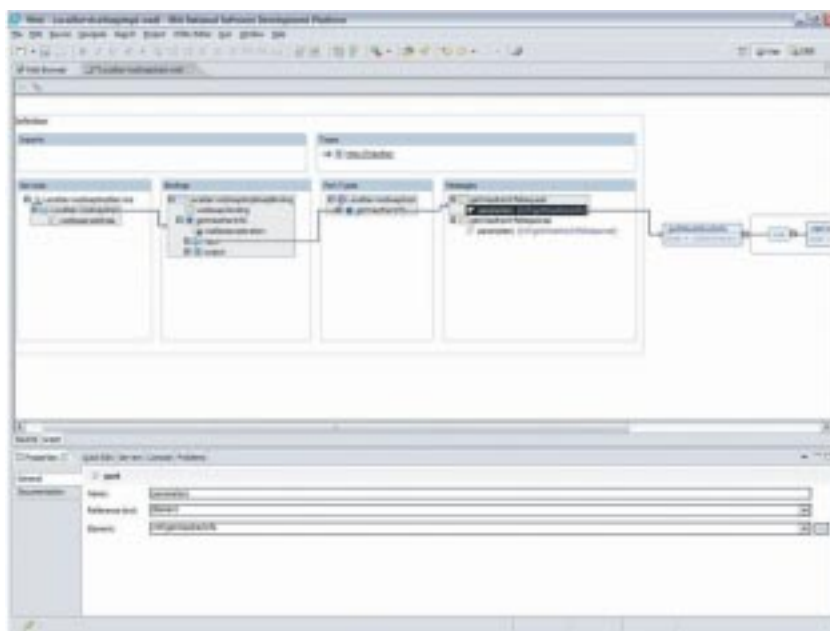


图 12. IBM Rational Application Developer 使得从现有资产 (例如, JavaBeans 和 EJB 组件) 创建新 Web Services 变得更加容易, 这是通过自动生成描述 Web Services 的 WSDL 文件、SOAP 部署描述符和用于测试 Web Services 的测试客户端来实现的。

使用 Rational Application Developer, 开发人员能够在 UDDI 注册中心注册服务可用性, 并创建服务的 WSDL 描述和一个供集成开发人员使用的 Enterprise Archive (EAR) 文件。一旦创建了这些资产, 开发人员就能够打包它们, 并将其存储在一个 RAS 兼容的存储库里, 例如基于 Rational ClearCase 的库。

测试人员: 测试服务

在开发人员交付可执行代码进行测试之前很长的时间里, 测试团队已经开始开发测试计划和测试案例。借助 IBM Rational 软件开发平台中的测试管理工具, 测试人员能够利用 Rational RequisitePro 的需求作为测试计划的输入资产, 并将这些需求链接到测试用例来验证它们。

一旦代码可用, 测试人员使用 **IBM Rational Functional Tester** 进行功能和回归测试, 确保服务功能与设计相符。测试人员还可以使用 **IBM Rational Performance Tester** 评估和验证 Web Services 性能, 确定服务在各种使用负载下性能表现如何。

集成开发人员: 组装和部署服务

集成开发人员的角色是编排构造的服务, 将其作为业务流程工作流的一部分。基于前期在 Web Sphere Business Modeler 中定义并导出为 BPEL 的总体业务流程模型, 集成开发人员通过将服务实现连接起来, 将整个工作流组装在一起。集成开发人员使用 WebSphere Integration Developer 导入、创建、发布和管理 BPEL 中描述的业务流程。

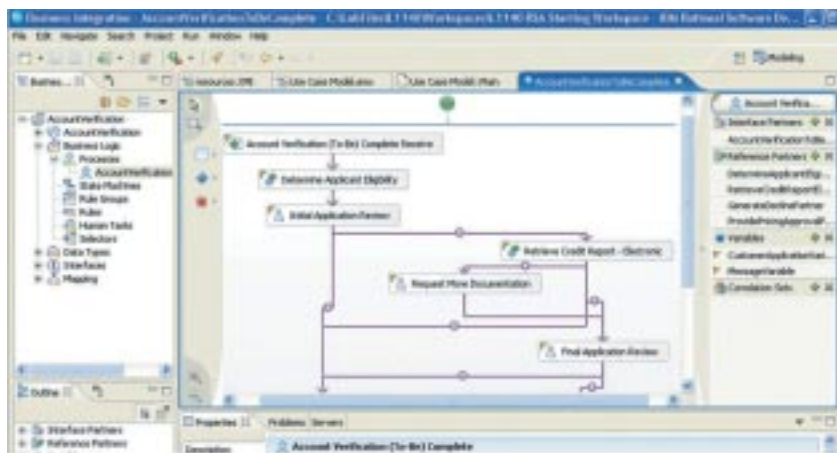


图13. WebSphere Integration Developer 中业务流程的 BPEL 实现.

集成开发人员还能够通过链接到提供 Web Services 的实现来开发业务流程。还可以包含一个在 UDDI 注册中心发现的服务作为业务流程中的服务提供者。集成开发人员还可以在测试前在业务流程中包含人员任务。在这一点上, 专业人员通常将业务流程部署到流程执行引擎中, 例如 WebSphere Process Server。

项目经理和团队: 管理资产生命周期

在整个开发生命周期中, 开发团队使用 IBM Rational ClearQuest 自动化和实施开发工作流。项目经理使用 Rational ClearQuest 分配任务和跟踪项目进度。开发人员和测试人员使用 Rational ClearQuest 作为集成的缺陷和变更请求管理工具。另外, Rational ClearQuest 能够被定制以支持资产认证, 即在资产被更广泛的用户使用之前, 复审、测试和验证资产的过程。

团队还使用 IBM Rational ClearCase 进行配置管理。因为 Rational ClearCase 和 Eclipse 的集成, 团队成员能够直接从 IDE 签出和签入项目资产, 从而极大地简化了并行开发。

为了推动 RAS 资产管理, 团队可以同时使用 RAS Repository for Workgroups 和 Rational ClearCase, 它们可以在 alphaWorks 上找到。

通过 RAS Repository for Workgroups 的 Web Services 接口, RAS Repository for Workgroups 支持下列功能:

- 使用 RAS 1.0 标存储库服务接口搜索和浏览存储库中的资产, 另一个增强的接口支持更复杂的查询

- 以多种格式检索资产信息，包括查看文档或反馈，导入完整的资产
- 将资产发布到存储库、创建和组织存储库中资产的逻辑视图、跟踪度量指标

项目经理和团队: 利用最佳实践和指南

通过配置具有 RUP Plug-in for SOA 和 RUP Plug-in for Web Sphere Business Modeler 的 IBM Rational Unified Process，整个开发团队任何时候都能够访问特定于 SOA 的过程指南。这些 RUP 插件为项目经理和团队提供捕获和模拟业务流程、设计和实施面向服务的应用程序的指南和最佳实践。

为了帮助确保项目的成功，解决典型软件开发问题的根本原因，团队要在单个服务的开发和组装复合式应用中应用 RUP 实践。

从 IBM 和 SOA 开始

过渡到 SOA 不需要企业马上对 IT 基础设施和开发流程执行一次完整的替换。过渡通常以循序渐进的方式发生。可以确定重新改写一个使用 SOA 的现有业务流程，或实现一个新的 SOA 应用程序。也可以从为软件开发周期的特定阶段采用面向 SOA 的开发方法和工具开始。图 14 显示的采用模型说明了采用 SOA 的四个级别。



图 14. IBM SOA 采用模型

在 Connect (连接) 级别上, 企业确保各种应用程序和系统之间有可靠而灵活的信息流。随着企业的发展, 它将在异构环境之间建立一个支持互操作性的集成框架: 移除建立 Web Services 和非 Web Services 所支持的集成架构和集成方法的障碍。下一步, 企业可以开始自动化业务和 IT 过程的编排, 让 IT 适应业务目标, 减少开支, 增加收入。最终, 企业可以用一个整体的方法优化过程, 通过让业务活动适应企业战略和运营目标、支持 IT 服务来转换和管理企业。

不管 SOA 采用处于哪个级别, IBM 定位于用一套完整的软件和服务产品来帮助启动和促进您的 SOA 项目。这些产品支持 SOA 生命周期的所有阶段, 包括服务的建模、组装、部署和管理, 从而建立有效的 SOA 解决方案。



结束语

灵活性对于寻求能快速响应变更的企业来说是一个基本需求, 这些变更包括出现的竞争威胁、遵循法规需求、合并、收购和不断演化的技术。灵活性就好像让业务流程 (即软件开发) 适应业务需求和优先级一样重要。

获得灵活性和让业务和IT目标更好地保持一致, 需要以较高层次的协同、准确性和清晰性来执行IT项目。面向服务解决方案的业务驱动的开发帮助企业创建能够真正满足当今企业需求的解决方案, 并且能够适应未来的需求变更。

IBM Rational软件开发平台在使企业能够创建面向服务的、能实现这些目标的解决方案方面发挥着重要作用。它包括了行业领先的工具、过程和产品中心管理、软件配置管理解决方案, 支持基于标准和新的编程模型, 从而能够自动化和加速面向服务的开发。

© Copyright 2005 IBM Corporation

IBM Software Group

Route 100

Somers, NY 10589

在美国印刷

09-05

版权所有

IBM、IBM 徽标、DB2、Lotus、Tivoli 和 Rational 是国际商业机器公司在美国和 / 或其他国家的商标。

Microsoft 是 Microsoft Corporation 在美国和 / 或其他国家的商标或注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标志。

本出版物中对 IBM 产品或服务的引用, 不代表它们可用于所有 IBM 运营的国家。

关于 IBM 未来方向或打算的声明仅代表 IBM 的发展目标, 如有变更, 恕不另行通知。所有信息均在“现有”基础上提供, 不提供任何类型的担保。

IBM 主页的网址为 ibm.com