在 Eclipse 中利用 Maven

充分使用这些工具

级别： 初级

Gilles Dodinet (gdodinet@karmicsoft.com), 研发工程师及顾问

2005 年 6 月 16 日

Maven 将作为一个普通的构建系统，被人们重新认识，并且它将超越 Java™ 技术。本文不打算成为一篇 Maven 教程，而是将 Maven 与其他技术进行比较，让您洞察 Maven 与 Eclipse 相适应的地方，以及如何使这些工具相互协作。

在过去几年中，Maven 已经不仅仅是 Java 世界的一个时髦话题。自 2001 年起，Maven 已经成为构建工具领域的先驱。最近几年，人们常拿它与 Ant 比较。因为 Maven 与 Ant 有一些显而易见的相似之处，所以拿它们俩作比较似乎是很自然的事。比如说，在两种情况下，XML 脚本都是可用的；两种工具都可以生产工件；它们还可以共享相同的分类法和概念，比如 项目（*project*）、目标（*target*）与 目标（*goal*），以及 依赖关系（*depends*） 和 先决条件（*prereqs*）。但它们实质上有很大的差别。Ant 只是一个 XML 脚本工具，而 Maven 是一个普通的构建工具，它关注的重点是一个叫做项目对象模型（POM）的概念。POM 暴露的是粗粒度的、面向构建的任务，这些任务被称为目标，它们提供了一些准则，帮助您开发构建方法和实现最佳实践。

在某些方面，Maven 属于软件工厂 工具家族（参阅 参考资料），尽管目前它仍在外围。更确切地说，诸如 Maven 之类的构建工具是软件工厂领域中必不可少的。

软件工厂

通过显著提高自动开发的级别，软件工厂提供了一个花费更少、更灵活的应用程序开发方法。根据 Software Factories Web 站点的说法，“软件工厂是一个软件产品线，它根据构建特殊种类的应用程序的方法，配置一些可扩展的开发工具.....提供打包的内容.....以及指导”（参阅 参考资料）。软件工厂涉及三个主要概念:

- 模式，提供了一些元数据，描述组成应用程序的不同工件的结构，以及它们是如何交互的。
- 一个或多个模板，提供了启动程序工具箱，以及构建应用程序所需的其他任何东西。
- 可扩展的开发环境，用于配置、定制和装配组件。

如今，这类工具已经逐渐引起了认识到它们的价值的那些工具制造人员、工具开发人员和软件编辑的注意。这些工具促进了软件开发过程的工业化，降低了投入市场的成本和时间，同时还提高了生产率，加快了对进化需求的反应。

仔细查看一下 Maven 的主要特性，您就会认识到 Maven 与软件工厂之间的相似之处：Maven 使用 POM 作为元数据来描述项目结构，并通过通用应用程序插件来获得可扩展的项目模板。因为 Maven 非常灵活并且是开放源码的，所以很容易推断和设想 Maven 是软件工厂平台中的一个核心组件。但是，还有另一种说法。上述类比并不完全，因为目前的 Maven 缺乏专用的开发环境，而这类环境有助于创建特定于域或特定于企业的插件和模板，并允许您轻松地配置项目或定制行为。

不过，Maven 的主要目标是标准化构建过程，并保证代码构建-测试-部署（CBTD）循环中的质量和易再现性（easy reproducibility）。它还可以制定度量标准，帮助您了解开发状态。CBTD 本体论在软件工程领域已经不再新鲜，但 Maven 可以使您标准化这种本体论，并通过抽象这种理论，将它想像成一个完整的实体。考虑到不断增长的项目的复杂性，标准化成为一种迫切需要。扩建 的概念，我们称之为元构建，因为其无可估量的价值以及保证下一级质量的特性，正逐渐被人们认识。持续集成（continuous integration）就是建立在这个概念的基础之上，但它也应用了在 IDE 上下文之外的地方进行构建的能力。

模糊的界限

使用过 Java 技术的人应该都听说过 Eclipse。2001 年年中的时候，Eclipse 推出了它的第一个版本，标志其成熟的是它为集成开发环境（IDE）提供了一个机会，特别是为 Java 开发人员（不严谨地说）提供了一个机会。Eclipse 是一种开放的、以语言为中心的平台，也可以将它用作教育性项目和研究项目的基础平台，其中一些平台捐赠给了 Eclipse 团体（请参阅 参考资料）。就像 Microsoft® 已经采用软件工厂方法一样，Eclipse 也开始转向模型驱动开发（MDD）方向，并且最近已经公布了一个新的项目提议 —— 模型驱动的开发集成（MDDi）。根据该提议，“Eclipse MDDi 项目专用于平台的实现……其设计目标是支持各种建模语言（统一建模语言或特定于域的语言）和模型驱动的技术。”

渐渐地，一些工具开始假定某些特性可以完全并且顺利地集成在一起，Maven 和 Eclipse（即使作为一个简单的 IDE）也不例外。因此，从构建的角度来看，二者似乎出现了重叠，如图 1 所示。

图 **1.** 扩建的概念

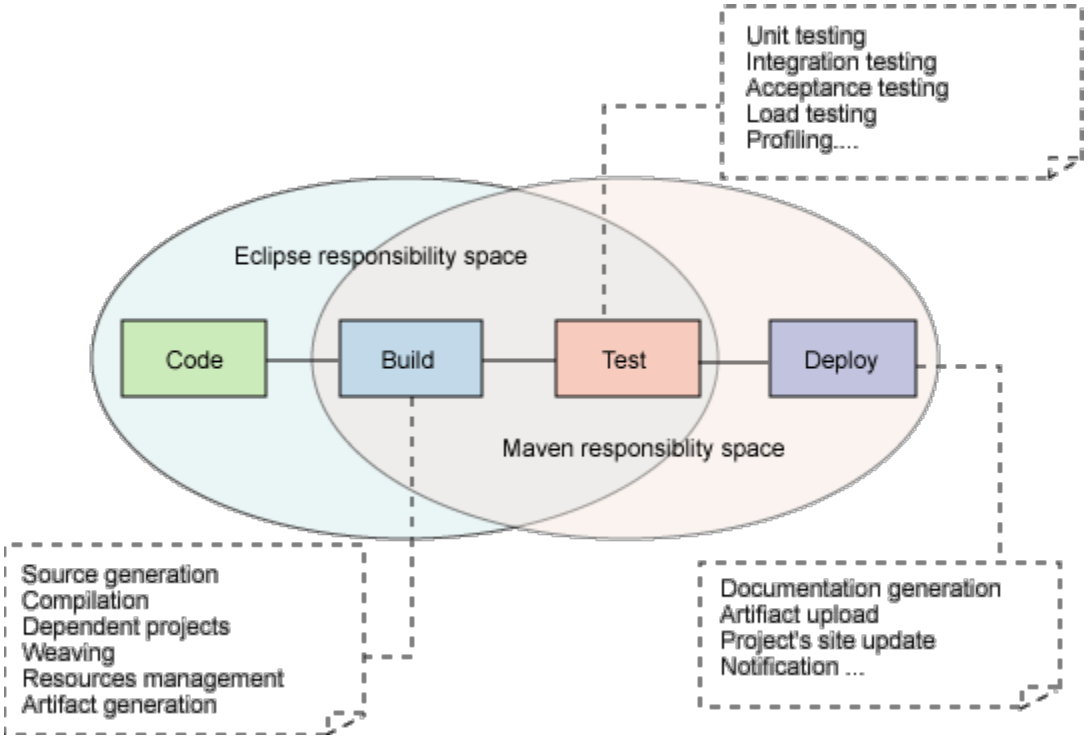

图 1 描述的实际上是以前讨论的扩建概念。正如以前定义的那样，整个扩建过程包括几个任务，同时还表示了一个元构建实例。任务可以是以下两种类型之一：原子任务是细粒度的，并且是上下文不明确的，这种任务的两个实例几乎是相同的；宏观任务是复合任务，它充当微观任务的容器。

构建，从扩展的意义上说，只处理宏观任务；原子任务的触发取决于配置。这意味着用户对系统有着较高层次的看法，这使得系统更易于维护和发展。

此外，Maven 和 Eclipse 都是开放的，并且很容易通过插件扩展它们，使其满足您的需要。但是，因为它们针对的受众不同，所以它们之间的相似性也到此为止：多数 Eclipse 最终用户是一些开发人员；而 Maven 主要针对的是一些构建管理人员。尽管如此，Maven 仍然是一个命令行工具。虽然图形用户界面（GUI）是按照 Jason Van Zyl（Maven 的制造者和架构师）指示的方向开发的，但 Maven 目前仍然没有帮助用户执行特殊任务的特定 GUI，比如创建或更新配置，或者只用一个鼠标单击发起构建。

上面描述的典型构建顺序在 Eclipse 中不像在 Maven 中那么顺利。Eclipse 的特性之一是开发 环境造成构建过程不连续，这要归因于一些人为因素：并不是每次成功编译之后都进行测试、所有测试没必要一次运行、可以跳过一些微观任务，等等。各种因素都会导致产生差异，这就是为什么开发人员每天至少必须运行一次完整的构建过程，以确信他们没有破坏什么的原因。

不过，因为 Eclipse 是一个可扩展平台，所以它受到许多用户社区的支持，这使它成为驻留 Maven 驱动的开发、允许开发人员和构建管理人员以某种简单的方式进行协作的理想之地。

回页首

将 **Maven** 集成到 **Eclipse** 中

Mevenide 是 Codehaus 主办的一个项目，旨在通过将 Maven 集成到 IDE 中，简化 Maven 的使用（参阅 参考资料）。现在，Borland Software 的 JBuilder、NetBeans 和 Eclipse 都受到支持。其他一些项目也部分地将 Maven 集成到 Eclipse 中，这样，就可以与 Mevenide（如 Maven Workshop）共享一些特性。除了增加 Maven 的易用性之外，为什么需要这种插件？

通过提供一些工具和视图，让您了解 Maven 隐藏的复杂性并改进团队环境中的协作，Mevenide for Eclipse 提高了生产率。从协作的角度来看，假如构建已经被 *Maven* 化了，那么最有用的特性就是 Eclipse 项目元数据与 Maven 元数据之间的双向同步。如果开发人员忘记在向 Eclipse 中添加一个依赖关系之后更新 POM，那么该怎么办？如果重构无法传播到 Maven，该怎么办？构建可能会中断，或者一些单元测试可能被拒绝，因此，真的需要使 Eclipse 元数据和 Maven 元数据保持同步。Mevenide 监听元数据的变化，并使您能够很容易地确定元数据不匹配的地方，这可以防止进行被严重破坏的构建。

但 POM 并不只是关于依赖关系和项目布局的。它还包含一些不用于结构上的项目管理信息，比如版本号、名称、ID 和源储存库的位置。Eclipse 元数据并不总是反映所有这些信息。因此，需要另一个编辑这些信息的方法。为此，Mevenide 提供了一个图形编辑器，该编辑器使 POM 的维护变得更容易，并允许您避开一些烦琐的、容易出错的原始 XML 手工编辑。每个 POM 语义部分都被表示为一个编辑器页，这增强了模型的整体可靠性。此外，为了最大限度地减少创建 POM 的无聊过程，Mevenide 提供了一个相当简单的、可扩展的 POM 模板机制。

阻止 Maven 在 Eclipse 之外的地方运行的能力是从类似 Mevenide 的插件中最容易获得的一项功能。您可以选择某些执行任务（用 Maven 的术语来讲是 目标），这些任务可以是通过 Maven 插件全局定义的任务，也可以是依赖于项目的任务，或者，您可以通过定义构建敏感的变量来定制构建。Eclipse 控制台上只显示了一些相关的选项（即在 IDE 的上下文中相关），对于其他任何插件，控制台上只输出构建日志。这一特性非常重要，因为它避免了在控制台与 Eclipse 之间不停地来回奔波。

Mevenide 还集成了其他许多帮助方法特性，但它们不是很重要。例如，您可以定义给定目标与文件模式之间的关系，这样，就可以根据工作空间增加的增量来激活那些任务。您还可以浏览工件储存库，或者根据名称搜索某个给定的工件（一个工件 就是一个构建结果，它可以是 JAR 文件、可执行文件或者是一个完整的 Web 站点）。然后，可以重定向到 Mevenide 站点，获得完整的特性列表。

不过，仍然还有一个问题：即使 Mevenide 增强了生产率和易用性，但您仍然必须认识到哪些工具可以在特殊环境中满足您的需要。通过 Maven Console 运行 Maven 会话可能非常耗时，所以，为了获得较高的生产率，在对会话进行编码期间，应该改为使用 JUnit 集成的 Eclipse 支持来运行测试，并依赖于 Eclipse 的内部编译器来生成可执行的文件。但在向源代码储存库提交任何东西之前，应该确保 POM 是同步的，并回滚所有不需要的 .classpath 或 .project 修改 —— 也就是说，假设这些文件都是在源代码控制之下，这可能是有争议的。这个话题在 Martin Van den Bemt 的 blog 中讨论过（参阅 参考资料）。

回页首

结束语

尽管 Maven 和 Eclipse 具有不同的特性，但从构建的角度看，它们在某些地方可能出现重叠。它们之间存在的对立似乎也相当多，但我们很容易克服这些，通过努力最终使它们相互协作，共创一片沃土。诸如 Mevenide 之类的工具可以使 Maven 和 Eclipse 顺利合作，并使它们保持同步，但您必须认识到，在某一个给定的开发阶段，哪种工具最能满足您的需要。

尽管 Maven 作为质量保证过程中的一个重要因素，正逐渐被人们所认可，但在将它集成到开发环境（尤其是 Eclipse）中，使 Maven 成为一等 IDE 公民这一点上，还有待提高。到那时，我们就可以考虑其他的 Maven 用例，比如说，更进一步地将它集成到全局开发过程中，以及像 Eclipse 这样的可扩展环境如何帮助实现这些集成。

本系列的下一部分将更详细地查看 Mevenide 的能力，以及如何使用该工具实现最佳实践，并从 Maven 和 Eclipse 中获得最大好处。

参考资料

- 您可以参阅本文在 developerWorks 全球站点上的 英文原文。
- Apache 中有许多参考文献和许多关于 Maven 入门的信息。那里还提供了一些最佳实践。
- *Software Factories: Assembling Applications with Patterns, Frameworks, Models and Tools* 一文由 Jack Greenfield 等人撰写，这篇文章深入查看了软件工厂，并描述了支持这些技术的一些关键元素。
- Software factories Web site 为软件工厂提供了理论基础。
- Eclipse.org 主办了 Eclipse 平台项目以及其他一些技术项目。
- First Experiments with a ModelWeaver 一文由 Jean Bezivin 等人撰写，文中提供了由 ATLAS 团队开发的 Model Weaver 的原型。
- Eclipse Model Driven Development Integration (MDDi) project 专用于实现提供应用模型驱动的开发（MDD）方法所需的那些工具的平台。
- 下载 Mevenide Eclipse 插件和其他有关的 Maven 插件。
- Maven Workshop 是一个简单的 Eclipse 插件项目，它与 Mevenide 共享一些特性。
- 在 Martin Van den Bemt 的 blog 中，讨论了团队环境中的 .project/.classpath 问题。
- “Eclipse 平台入门”（developerWorks, 2002 年 11 月）提供了 Eclipse 的历史和概述，其中包括如何安装 Eclipse 及其插件的细节。
- *Eclipse in Action: A Guide for Java Developers*（Independent Publishers Group, 2003）是使用 Eclipse 的 Java 开发人员的必读之物。
- 从 DB®、Lotus®、Rational®、Tivoli® 和 WebSphere® 获得一些评价较高的产品，然后开始构建应用程序，并在 IBM 中间件上部署它们。请选择 Linux® 或 Windows® 版本的免费的 Software Evaluation Kit (SEK)。
- 请参阅 developerWorks 的 开放源码专区，从中获得大量 how-to 信息、工具和项目更新，帮助您使用开放源码技术进行开发，并将这些用于 IBM 的产品。
- 使用 IBM 试用软件 改进您的下一个开放源码项目，可以通过下载或从 DVD 上获得这些软件。
- 在 Developer Bookstore 的开放源码专区，可以找到数百本 打折出售的关于开放源码主题 的书籍，其中包括许多 关于 Eclipse 的书籍。
- 通过参与 developerWorks blogs 加入 developerWorks 社区。

关于作者

Gilles Dodinet 曾做过几年的 Java 2 Platform, Enterprise Edition（J2EE）工程师。自 2004 年起，他成为了 KarmicSoft 的一名研发顾问，从事 .NET 组件装配工具方面的研究。他是一名经验丰富的 Eclipse 开发人员，并且参与开发了 Mevenide。Gilles Dodinet 非常感谢 Michel Zam 和 Milos Kleint 对本文的审查。

对本文的评价

- 太差！ (1)
- 需提高 (2)
- 一般；尚可 (3)
- 好文章 (4)
- 真棒！(5)

建议？

反馈意见

回页首