JAVA 技术丛书

精通XMI

——使用XMI、XML和UML进行Java编程

Mastering XMI

Java Programming with XMI, XML, and UML

Timothy J. Grose
[美] Gary C. Doney 著
Stephen A. Brodsky

徐 强 金艳红 等译

WILEY

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

Java 技术丛书

精通 XMI

——使用 XMI、XML 和 UML 进行 Java 编程

Mastering XMI

Java Programming with XMI, XML, and UML

Timothy J. Grose
[美] Gary C. Doney 著
Stephen A. Brodsky

徐 强 金艳红 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书围绕着最新的XML高级技术——XML元数据交换（XMI），讲解了如何使用XMI、XML和UML进行Java编程。全书分为两部分，共11章。作者首先介绍了基本的XML和UML概念，从而帮助读者更好地理解XMI。书中重点讨论了如何创建XMI处理和XMI模型，特别是从XML文档、DTD和模式中反转工程模型；介绍了使用标准XML API（DOM和SAX）与框架来创建和读取XMI文档；最后，本书还解释了XMI与模型驱动体系结构（MDA）的协作，并且讨论了在IBM的WebSphere Studio Application Developer中的XMI应用。本书以大量的Java实例为基础，可以使读者更好地学习与掌握XMI、XML、UML等相关技术。随书附带的光盘上除了包含书中的一些完整实例之外，还提供了XMI框架、解析器以及其他的软件工具。

本书是一本实用的XMI参考手册，可供从事软件开发的相关技术人员和编程爱好者使用。

版权贸易合同登记号　图字：01-2002-5161

图书在版编目（CIP）数据

精通XMI——使用XMI、XML和UML进行Java编程/（美）格罗斯（Grose T. J.）等著；
徐强等译. -北京：电子工业出版社，2004.2
（Java技术丛书）
书名原文：Mastering XMI: Java Programming with XMI, XML, and UML
ISBN 7-5053-9615-3

Ⅰ.精… Ⅱ.①格… ②徐… Ⅲ.①可扩充语言，XMI、XML－程序设计 ②面向对象语言，UML－程序设计 ③JAVA语言－程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2004）第003798号

责任编辑：冯小贝
印　　刷：北京兴华印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：23.75　字数：532千字
印　　次：2004年2月第1次印刷
定　　价：39.00元（附光盘1张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

译 者 序

随着 Internet 的不断壮大以及网络技术的不断发展，可扩展标记语言（Extensible Markup Language，XML）与 Java 正受到越来越多的关注。XML 的简单性、严格性以及良好的数据传输性，将使其成为下一代网络技术发展的核心；而 Java 的可移植性、分布性以及面向对象的特性，也使其在网站建设和网络维护中担当着重要的角色。如果能将这两者很好地结合起来，那么我们就会迎来一个全新的网络世界。但是，在 XML 中表示对象还是一件比较困难的工作，必须依靠相应的技术在 XML 和应用对象之间进行转换。随之而来，一种新型的 XML 技术诞生了，这就是 XML 元数据交换（XML Metadata Interchange，XMI），它可以将 XML 文档直接映射为 Java 定义的对象，或与其他的软件工具交换这些对象，并且在 XML 应用程序中实现建模。

本书正是围绕这一最新的 XML 高级技术（XMI），向读者讲解了使用 XMI、XML 与 UML 进行 Java 编程，并讨论了如何在一些软件开发工具中应用 XMI。全书共分为两部分，首先介绍了 XMI 的基本概念与相关的 XML、UML 技术。然后重点讲解了怎样使用 XMI，利用标准 XML API（DOM 和 SAX）与框架（Framework）来创建和读取 XMI 文档，以及从 XML 文档、DTD 和模式中反转工程模型。本书还特别介绍了一种新型的软件开发方法，即模型驱动体系结构（Model Driven Architecture，MDA），并示例了 XMI 在 IBM 的 WebSphere Studio Application Developer 中的应用。

本书以大量的 Java 实例为基础，可以使读者更好地理解 XMI、XML、UML 等相关技术。随书附带的光盘上除了包含了书中的一些完整实例之外，还提供了 XMI 框架、解析器以及其他的软件工具。这将帮助读者更好地学习 XMI 并掌握这项技术。

本书主要由徐强、金艳红组织翻译工作。参加本书翻译的人员还有常丽莉、邱晓彬、毕波、徐申、崔蔚、刘丽丽、李奕、杨长宇、赵日升等。由于译者的水平有限，书中定有不少疏漏之处，敬请读者批评指正。在此，译者向所有热情的读者致敬！

前　言

我们现代日常生活中的一些设施由于标准而变得可以实现。今天，你可能已经拨打了一个电话，从朋友或者商业伙伴那里收到或发出了一份传真，或者阅读了一些Email，其中可能包括一些需要某种程序才能打开的附件。也许你想要欣赏一些音乐CD，那么可以选择不同的方式进行播放，可以是家里的视听设备、车载音响或CD随身听，甚至通过计算机的光驱。在音像商店或通过网络购买CD时，不需要担心无法在上述四种设备中播放。这是因为CD上的音乐信息是以标准化的格式编写的。类似于音乐CD，许多经过配置的现代化设施为人们带来了很多益处并且可以选择使用，这是因为它们被设计并构建成与可接受的工业标准一起工作。

XMI是一种使用了XML来说明面向对象的信息的标准，它以可扩展标记语言（Extensible Markup Language，XML）元数据交换为标准。对象管理组（Object Management Group，OMG）是一个工业领域的协议，XMI 1.0于1999年2月采用了该协议，OMG增强了实现不同系统之间的交互操作性的标准。为了响应OMG的公共要求（Request for Publication，RFP），将XMI 1.0开发成一个以流水线为基础的标准来说明面向对象的信息，并且可以由29个工业领先的公司所支持。因为XMI具有表示多种格式的面向对象信息的功能，支持XMI的软件可以在Java应用程序、网络、XML及不同种类的模型之间提供整合。

XMI已经发展到遵循OMG的开发方法和过程。正因为如此，XMI的4个版本已经符合了OMG的标准。除了支持XML名字空间的XMI 1.0之外，XMI 1.1已经在2000年4月被采用。XMI 2.0是最新的版本，本书正是以这个版本为基础的。XMI 1.2和XMI 2.0在2001年11月已被OMG采用。XMI 2.0为XML模式（schema）提供支持，同时还支持从模式、文档和DTD中反转工程。最后，在解决一些关键的问题中，仍有一些关于早期XMI版本的正在进行的工作。

根据读者的编程经验，可能已经熟悉了对象的概念。如果读者已经使用过Java，那么就知道对象（伴随着用来描述它们的类）是这种语言的基本单元。当Java程序运行的时候，Java虚拟机就创建了对象，这个对象执行已经编写的方法中的编程指令。如果读者已经使用过可视化建模工具，那么可能用过统一建模语言（Unified Modeling Language，UML）或者是相似的方法创建了类和对象。如果读者编写过C++或者Smalltalk等语言的程序，那么可能已经处理过对象。

如果读者已经了解了对象，那么就知道有许多不同的方法来表示它们。我们使用过的工具可能有一些特殊的文件格式，它们用来保存有关对象的模型或者是实例。对于使用它们的工具来说，这些格式可能非常合适，读者从来没有想过需要有其他的方式来说明正在使用的对象；

也可能想当然地认为只要需要对象的应用程序能够访问它们，那么就不需要使用不同的方法来表示它们。

尽管如此，今天的计算机世界日新月异。与以前不同，交互操作性变成了关键需求。通信技术的进步以及网络的快速发展对于一起工作的系统已经提出了更多的要求，正如提供了更多的机会一样。继而造成的结果是，需要共享这些系统所依赖的数据，并且共享以最初开发系统时没有考虑的方式进行。虽然在一些情况中，可以从一个应用程序到另一个建立定制的软件桥梁，但使得面向对象的数据能够共享的惟一有效方式与表示数据的标准方法有关。

让我们考虑一个来自日常生活中的假设情况，其中表明了标准如何扩大了做出选择的范围。假设你决定购买一个崭新的真空吸尘器。在查看了当地几个商店中可以买到的吸尘器之后，还没有决定要买哪一个。后来看到了一个广告，这是一个新推出的真空吸尘器，即一种新型的、革命性的Vac-o-matic，这种吸尘器只以邮购的方式出售。正如这个广告所说明的一样，Vac-o-matic只由Vactron公司制造。Vac-o-matic的特殊之处是其对于革命性的声音流技术的使用。这个广告说明了Vac-o-matic具有新的功效，当这个吸尘器清洁地毯的时候，它通过专利的sonic强力旋转喷嘴（dirt blaster）而发射出特殊的声音。Vactron公司的研究者们宣称，sonic强力旋转喷嘴将震动地毯纤维中的灰尘而使其松动，因此Vac-o-matic能够比传统的真空吸尘器多收集10%的灰尘。同样，Vactron公司宣称，来自早期使用者的非官方报道已表明，这种特殊的声音对于家庭宠物以及小孩子没有很大的影响。

你立刻确定Vac-o-matic是最适合的，并且拨打了广告中所显示的免费电话。在与Vactron公司一位名叫Carol的职员通话之后，她表示在几天之内会将新的Vac-o-matic送达。正如事先承诺的一样，这台吸尘器附带10个免费的过滤包。用户指南说明，应当在每次使用Vac-o-matic之后更换过滤包，从而避免损坏相应的部件。

在第一次使用Vac-o-matic的时候，它非常好用，并且把地毯清洗得非常好。但是，你注意到家里的狗受到了Vac-o-matic产生的声音的影响。在每次打开Vac-o-matic的时候，狗就开始叫。你认为Vac-o-matic可能出什么毛病了，于是便给Vactron公司打了电话。Carol接听了电话并且听取了意见。为了检查Vac-o-matic是否有问题，Carol请你打开吸尘器并将电话的听筒放在离Vac-o-matic很近的地方，以便她可以听到它的声音。令人欣慰的是，Carol告诉你Vac-o-matic是没有问题的，她说问题一定是出在狗身上。不过，Carol告诉你，其他提出相似问题的顾客已经发现，在吸尘的时候将狗放置在屋外它就不叫。你可以考虑一下，然后决定需要怎样做。

两个月以后，已经用完了Vac-o-matic附赠的过滤包，你决定从商店中再挑选一些。在当地的三家商店里寻找之后，你才发现其中没有一家出售用于Vac-o-matic的替换包。最后到达的一家商店的销售经理告诉你，用于Vac-o-matic的过滤包与商店出售的其他过滤包不同。另外，用于Vac-o-matic的过滤包不能用于市场上的其他任何真空吸尘器。这位经理同时说明了

因为卖出的Vac-o-matic吸尘器很少，所以用于这种吸尘器的过滤包是不值得进货的。他建议打电话给Vactron公司，询问他们是否可以通过邮购的方式提供用于Vac-o-matic的过滤包。

当你回到家以后，立即打电话给Vactron公司并再次与Carol通了电话。Carol解释说他们在包装盒里提供了一个指定的邮购订购格式，Vac-o-matic就是用这个盒子运来的。虽然不能通过电话定购过滤包，但是可以按照这种方式来订购它们。在结束了通话之后，你检查了Vac-o-matic的包装盒，并且发现过滤包的订购格式就写在盒子底部。替换过滤包一次必须至少订购100个，同时价格比商场里所看到的过滤包要贵很多，并且需要等待6～8个星期才能送到。因为没有专门的过滤包，就不能够使用Vac-o-matic，所以你决定从Vactron公司订购它们。

大约两个月以后，替换的过滤包送达了，你再一次开始使用Vac-o-matic。但是，这一次却发现有一种来自发动机的奇怪声音。于是你打电话给Vactron公司，并且再次与Carol小姐进行了通话。Carol请你描述了这种声音，她解释了最有可能的原因是sonic强力旋转喷嘴中的旋涡流已经不起作用了。Carol告诉你，惟一的解决办法是将Vac-o-matic送回公司，以便请他们的某位修理人员进行检查。Carol同时也告诉你，将Vac-o-matic送到当地的修理店是没有意义的，这是因为普通修理店只提供标准的工具，需要Vactron公司才有的特殊的制造工具来修理Vac-o-matic的sonic强力旋转喷嘴。

这一次，你询问Carol是否能够办理退货，因为现在已经发现Vac-o-matic的确不太合适。虽然它可能是一个很高效的吸尘器，但是它与传统的吸尘器区别太大，以至于很难对其进行保养和修理。Carol说她理解你的感受，但是因为现在已经过了保修期，所以Vactron公司是不能办理退货的。不过，Carol表示Vactron公司允许你用Vac-o-matic换购该公司最新型的真空吸尘器——Vac-o-magic。Carol解释说，Vac-o-matic的sonic强力旋转喷嘴中的旋涡流通常都会失去作用，这是因为它使用的是twirl-a-whirl技术。但是，在Vac-o-magic的sonic强力旋转喷嘴中，由于使用了最新的、专有的spin-a-tron技术，因此更加耐用。Carol还告诉你，只需要支付299.99美元就可以用原来的Vac-o-matic换购最新的Vac-o-magic。

在经过思考之后，你告诉Carol不打算换购最新的吸尘器。相反，你将要选择一家知名度更高的真空吸尘器。接着，你来到了当地的商店，在这里可以实际试用真空吸尘器。你发现了一个自己非常满意的吸尘器，同时也确信商店中出售用于这种吸尘器的替换过滤包。令你惊奇的是，有许多制造商都在生产真空吸尘器的过滤包。这是因为过滤包是以一定标准为基础的，它能够用于其他的真空吸尘器。现在，你已经有了一个关于过滤包制造商的选择，同时确定了购买的数量。另外，如果需要更多的过滤包，则可以从附近的商店进行购买。而且，如果真空吸尘器出现了问题，可以到当地几个修理店进行修理，这是因为不需要专门的工具来打开和修理这种吸尘器。所有这些都是可能的，这是因为你从一个以专有技术为基础的机器转移到了以广泛采用的标准为基础的机器。

正如这个假设的情节所描述的一样，局限在一个公司的专有技术之中的选择，将会对用户产生深刻的影响，尽管使用了开放标准的这种选择具有很多优点。通过去除将顾客封锁在市场

中的某个公司的障碍，开放标准对相关的领域进行了分层。当使用开放技术的时候，所有的供应商都可以免费创建新的工具和应用程序，它们与已经存在的产品一起工作，因此可以创造更大的价值。结果顾客可以享有更多的选择，并且可以选择最适合他们需要的工具、系统和产品。同样，XMI作为一个广泛采用的工业标准，它为使用对象技术的用户带来了同样的好处。XMI可以促进数据的交换、增强应用程序的集成，并且可以实现程序交互性的某种层次，这是专有数据格式无法实现的。

虽然XMI中的M代表元数据，但是可以在应用程序中使用XMI，即使用户可能并不熟悉元数据的相关术语，或者认为应用程序没有使用元数据。元数据和数据之间的区别对有些应用程序是有用的，尤其是那些涉及到的数据具有多重抽象层次的应用程序。但是对于所处理的数据只在一个抽象层次上的应用程序，这种区别是不重要的。即使应用程序处理在同一抽象层次上的数据，使用XMI仍然可以获益。在本书中，我们将讨论XMI和建模是怎样成为一个全面的模型驱动体系结构的一部分，这个体系结构集成了Java、软件模型、XML和Web。

本书概览

读者选择阅读这本书的最有可能的原因，就是试图确定XMI是否适合正在开发的技术，以及怎样开始使用XMI。读者想要知道XMI的具体内容，以及怎样开始编写创建和读取XMI文件的应用程序。我们相信，通过查看实例和进行练习，将会更好地学习XMI。正因为如此，本书是一本非常实用的参考手册。其中包含了采用Java编写的程序实例，这些程序表明了如何创建和读取XMI文件。

我们遵循的一般形式是首先介绍一个概念，接着详细对其进行解释，然后给出说明这个概念的程序、模型或者XMI信息的实例。对于书中的一些大型程序，我们设法采用分步的方式，以便更容易理解这些程序。另外，在整个程序源代码中给出了注释，可以从中参考其他章节讨论的概念或算法。我们感到这是理解概念的最好方法，并且有助于将知识转化为实践。

除了讨论XMI之外，我们提供了相关技术的一些信息。这些相关技术是UML、OMG的元对象工具（Meta Object Facility，MOF）、XML（包括DTD和模式），以及两个XML应用程序编程接口（Application Programming Interface，API）——文件对象模型（Document Object Model，DOM）和用于XML的简单API（Simple API for XML，SAX）。因为完整讨论这些概念需要很长的篇幅，所以我们并不打算涉及这些主题的所有细节。但是本书提供了足够的背景知识和细节，以便读者理解这些技术是如何与XMI相关的，并且考虑我们给出的实例。为了帮助读者更加深入地理解这些主题，我们提供了相关文件和规范的一个参考清单，其中可以找到更多的信息。

XMI是以OMG的MOF为基础的。MOF提供了一种层次结构，因此可以在较高的抽象层次上表示信息。目前，OMG正在努力结合UML 2.0和MOF 2.0。这样产生的结果是，使用UML模型和使用MOF模型是一样的，这是有可能的。因为对于我们给出的模型来说，UML和MOF的当前功能足够帮助人们理解XMI。并且因为许多人已经知道了UML，或者已经使用了UML工具，所以我们将在这本书中使用UML实例来解释XMI。在第10章中，我们解释了MOF、模型驱动体系结构（Model Driven Architecture，MDA），以及UML如何适应这些技术和概念。因为本书提供的实例并不依赖于MOF层次结构中的特定层次，所以对于我们使用UML来说是足够的。

最后，正如我们在前面提到的一样，遵循着OMG的开发方法，XMI取得了一定发展。这本书是以最新版本XMI 2.0为基础的。虽然我们并没有专门介绍XMI 1.0、XMI 1.1和XMI 1.2，但是书中提供了许多的实例，它们解释了XMI的当前版本和以前版本之间的区别。将来，OMG还会继续修改XMI。虽然本书的目的是反映XMI 2.0在当前格式下的规范，但是我们期望伴随着XML、MOF和UML技术的发展，XMI规范也能不断地完善。

本书的组织结构

这本书包含以下两个部分：

- 第一部分：XMI概述
- 第二部分：使用XMI

第一部分提供了XMI的概述。我们也给出了关于XML和UML的介绍，读者需要知道这两种技术来理解XMI以及本书所使用的模型。第一部分包含了以下3章：

- **第1章——XMI：XML中的对象表示。**本章解释了XMI提供了一个标准方法，用于在XML中表示对象。在这一章中，通过说明XML中表示对象的标准方法如何消除了模糊性（没有这些标准时产生的），我们给出了使用XMI的原因。同时，本章解释了XMI如何促进了操作对象的不同工具的集成，同时也讨论了利用XMI标准的其他好处。
- **第2章——相关标准：XML和UML。**本章介绍了XML和UML中的一些基本概念。为了理解XMI和本书中给出的实例，需要读者了解这些基本概念。涉及到XML的概念包括元素、属性、名字空间、DTD和模式。对于UML，我们所涉及的基本概念包括类、对象、属性、联合、关联点以及软件包。如果读者已经熟悉了XML或者UML，那么可以跳过这一章的相关章节。
- **第3章——XMI的概念。**本章详细讨论了XMI 2.0中面向对象的信息。以我们在第2章中对UML的介绍为基础，这一章解释了根据XMI标准，如何在XML的UML对象模

型中表示概念。同时，我们提出了与创建XMI模式有关的问题。最后，本章检查了由XMI模型提供的元素和属性，以及在XMI文档中，怎样利用它们来说明附加信息。这些内容包含了扩展、差异（增加、删除和替换）以及有关文档中的数据的细节，例如数据所基于的模型。

根据第一部分对于XMI的理解，我们在第二部分学会如何在自己的Java应用程序中使用XMI。在第二部分中，我们介绍了一个能够使用的一般性的开发过程；在开发使用了XMI的模型时，应当遵循的详细指导；以及贯穿一些编程实例的步骤，这些编程实例使用了为XML和XMI设计的软件。第二部分包括以下8章：

- **第4章——创建XMI处理。**本章给出了一个含有5个步骤的开发方法，如果决定在开发项目中使用XMI，那么我们建议遵循这个步骤。这个处理过程表示在一个通用格式中，可以按照需要进行裁剪。为了让读者更好地理解这个处理过程，通过为一个示例应用程序开发模型，我们实现了最终的程序。然后，我们为已经开发的模型生成了一个XMI模式。另外，我们给出了Java代码，它是从模型中生成的。
- **第5章——创建XMI模型。**本章从涉及到UML建模和XMI的问题开始讨论。接下来，我们解释了一些常见的算法，它们用于从XML文档、DTD和模式中反转工程UML模型。
- **第6章——使用标准XML API来创建和读取简单的XMI文档。**本章涉及到如何使用一套标准的XML接口来阅读和编写XMI文件，这两个标准的XML接口是DOM和SAX。本章还介绍了一个汽车租赁代理模型，我们在这一章以及后面的一些章节中将使用这个模型。如果对掌握如何实现标准XML接口不感兴趣，而是喜欢学习专门为处理XMI设计的软件，那么可以跳过相关的章节。但是，读者应当阅读本章开始部分给出的汽车租赁代理模型，以便能够了解一些利用了这种模型的实例，这些实例将在本书的后面出现。
- **第7章——使用XMI框架来创建和读取简单的XMI文档。**本章介绍了书中涉及的用来执行XMI的软件。在这一章的开始部分，我们介绍了Java对象桥接（Java Object Bridge，JOB）。我们说明了如何使用这个非常简单的接口来读取和编写包含了Java编程对象表示的XMI文件。接下来，我们介绍了XMI框架，它提供了一个用来执行XMI的API。我们说明了如何使用API框架来读取和编写XMI文件，以及在框架中如何表示UML模型。
- **第8章——使用XMI框架来创建和读取高级的XMI文档。**本章以我们在第7章中提供的XMI框架为基础。在这一章中，我们讨论了一些编程实例，这些实例利用了一些更加高级的XMI特性。我们所涉及到的主题包括XML名字空间、XMI扩展、ZIP文件的使用以及交叉文件引用。

- **第 9 章——XMI 模式。**本章解释了如何使用 XMI 框架来为一个模型产生一个模式，以及在使用框架载入一个 XMI 文档的时候，如何进行验证。这一章也详细描述了错误的各种类型，即对于产生模式而影响验证而言，验证检测并解释了如何选择不同的XMI选项。
- **第 10 章——模型驱动体系结构（MDA）和 XMI。**本章介绍了开发软件的 MDA 方法。在这一章的开始部分，我们解释了 MDA 的目标，并且给出了 MOF 的分类学的概述，MOF 的分类学是为了说明信息抽象的不同层次。接下来，我们使用建模方法而不是 UML（FCM）开发了一个实例。本章解释了如何在 XMI 中表示这个实例，以便使 MDA 方法的关键——信息共享成为可能。
- **第 11 章——XMI 使用实例：WebSphere Studio Application Developer。**本章考察了如何在 IBM 的 WebSphere Studio Application Developer 中使用 XMI。在这一章中，我们首先讨论了如何在工具中使用 XMI，这个工具可以指定从一个 XML DTD 到另一个 XML DTD 或者从一个 XML 模式到另一个 XML 模式的映射。然后，本章解释了如何在一个涉及到企业级 JavaBean（EJB）的实例中使用 XMI。

本书还包括一个附录，其中提供了有关 XMI 框架的更多信息，这些信息是对第 7 章、第 8 章、第 9 章中所提供的框架内容的补充。

本书的读者对象

本书主要是为Java软件开发人员编写的，他们正在考虑使用XMI。因为XMI是基于XML的，而且本书中的大多数实例是基于UML的，所以使用过XML或者UML将会帮助理解本书的内容。尽管如此，我们提供了有关 XML 和 UML 特性的介绍，这些特性对于理解本书的内容也是有帮助的。因此，了解 XML 和 UML 的知识不是必需的。如果读者认为对这两个主题相当熟悉，那么可以跳过相关的章节，只阅读那些还没有掌握的内容。

我们给出的程序实例都是采用Java语言编写的。为了理解这些实例，读者应当具备一些利用 Java 进行编程的经验，或者至少要熟悉这门语言的有关语法，但是不需要掌握 Java 的中高级特性。只要理解了一些概念，例如类、接口、流程控制结构和继承机制，就可以理解我们给出的所有实例。尽管在一些实例中，可能会涉及到一些更加高级的Java概念，例如串行化和反射，但是也不需要详细了解这些概念。如果读者具备一些编程语言的经验，并且这门语言（例如C++）的语法与Java的语法相类似，那么就能理解实例的一般流程，不需要掌握更多的Java知识就可以在自己的系统上运行这些实例。

如果读者不是一名软件开发人员，但是却对以下两方面很感兴趣，即 XMI 是关于什么内容的，以及个人或者组织是否应当考虑使用XMI，那么通过阅读本书就可以获得满意的答案。本书的前两章介绍了 XMI、XML 和 UML 的基本概念，同时对于 XMI 的具体含义以及如何使

用XMI也给出了一个总体描述。另外，第10章介绍了MDA，这是一个利用XMI的新的软件开发方法；第11章讨论了在WebSphere Studio Application Developer中如何使用XMI。上述内容都会对读者有所帮助。

本书一共包含11章。虽然我们希望读者阅读所有的章节，但是根据图1给出的各章之间的路径图，可以确定最适合读者的阅读方式。在这幅路径图中显示了本书的一些章节，我们认为它们是可以选择的阅读材料。根据读者的知识背景或需要，可以选择跳过这些章节或是大致浏览这些章节。例如，第2章包含了为相关读者提供的XML和UML的介绍，这些读者可能不了解这些概念或者只是有限地掌握了这些技术。如果读者已经掌握了这两个领域的相关内容，那么可以跳过这些章节。

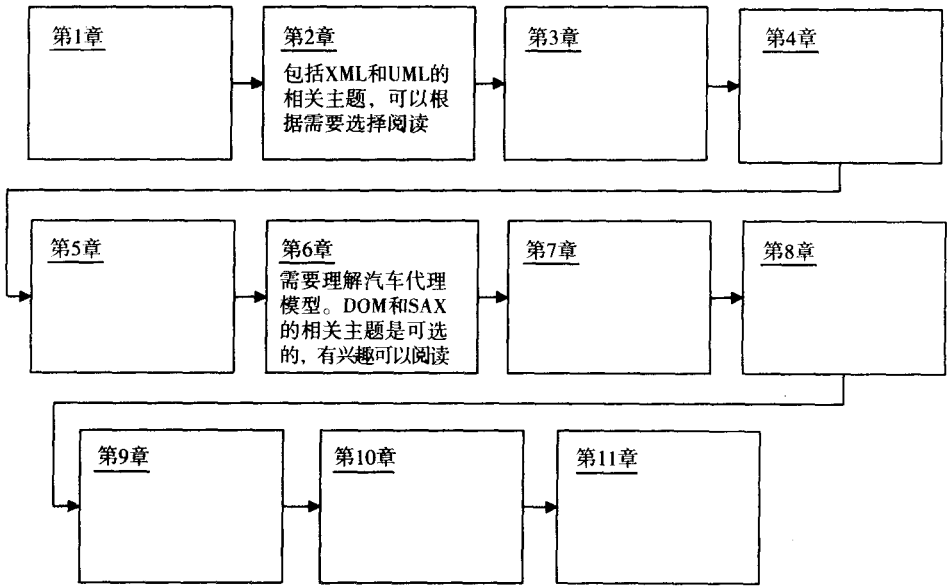

图1　本书的阅读路径图

如果读者只是想要大概了解XMI，并且不打算使用XMI进行编程，那么可以遵循图2给出的概览路径图。按照这种路径，将使读者在不知道如何在XMI中表示信息或者如何使用XMI编写程序的情况下，就能理解XMI是很有价值的。

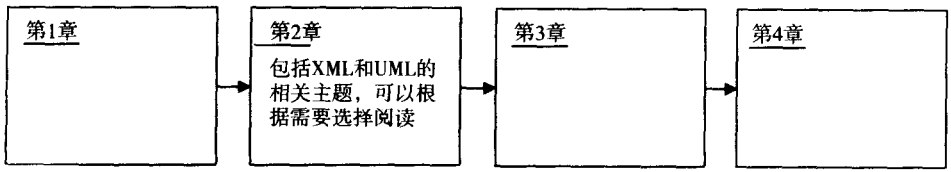

图2　本书的阅读路径略图

光盘的内容

本书附带一张光盘，其中包含以下内容：

- 一个特性完全的关于 WebSphere Studio Application Developer 的试用版；Java 2 平台企业版（Java 2 Platform Enterprise Edition，J2EE），这是一个适应服务器端的 Java 工具，关注于关系数据库、Web 站点建设、Web 服务以及 EJB 的开发、部署和建模。同时还包括 IBM 的 WebSphere Studio Site Developer 的所有功能。
- 书中出现的一些程序实例，包括第 6 章、第 7 章、第 8 章和第 9 章中的例子。
- XMI 框架，它提供了用于读取和编写 XMI 文件的 API。其中给出了第 7 章、第 8 章和第 9 章中使用 XMI 框架的例子。
- JOB，它在 Java 程序中创建的对象和 XMI 之间进行转换。我们给出了第 7 章中有关 JOB 的例子。
- 对应于 IBM 的 XML4J 解析器（包括了 Apache Software Foundation 的 Xerces Java XML Parser 的所有功能）版本 3.2.1 的 XML4J-J-bin.3.2.1.zip 文件。XML4J 解析器用来运行第 6 章、第 7 章、第 8 章和第 9 章中出现的例子。

可以查看光盘根目录下的文件 Readme.htm，其中给出了相关软件的安装指南。

光盘使用的相关要求

下面的内容给出了运行光盘上的程序的相关要求。注意，这些要求对于光盘上的所有软件并不是相同的。一般情况下，运行 IBM 的 WebSphere Studio Application Developer 的要求要大于其他软件的要求。因此，如果系统符合了 WebSphere Studio Application Developer 的要求，那么也能运行光盘上的其他程序。

接下来是运行 XMI 框架、JOB 以及这本书中的程序实例的要求。对于运行光盘上所包含的 XML4J 解析器，这些要求也同样是足够的：

- 最低的硬件设备要求至少有相当于 Pentium 处理器的 CPU（主频为 90 MHz），以及 64 MB 内存。推荐的硬件设备要相当于或者高于 Pentium II（CPU 主频为 300 MHz），以及 128 MB 内存。
- Windows 2000、Windows ME、Windows 98 或者是 Windows NT 4.0，Service Pack 为 6a 或者更高。
- JDK 1.2.2 或者更高版本。

WebSphere Studio Application Developer 具有下面这些软件及硬件的要求：

- Windows 2000、Windows ME、Windows 98 或者是 Windows NT 4.0，Service Pack 为 6a 或者更高。
- Microsoft Internet Explorer 5.5 或者更高版本。
- 已经安装或者配置了 TCP/IP 协议。
- 鼠标或者是可以替代的指示设备。
- Pentium II 或者是推荐的更高的版本。
- SVGA（800 × 600）显示效果或者更高（推荐为 1024 × 768）。
- 内存最小为 256 MB。
- 硬盘空间要求：最小为 400 MB（基于 NT 文件系统[NTFS]，实际上文件分配表[FAT]的硬盘空间依赖于硬盘驱动的尺寸和分区）。

关于安装、部署平台、所支持的软件以及相关主题的附加信息将包含在 WebSphere Studio Application Developer 提供的文件中。光盘上的 Readme.htm 文件描述了怎样获得这些信息。

目　录

第一部分　XMI 概述

第二部分　使用 XMI

第一部分

XMI 概述