

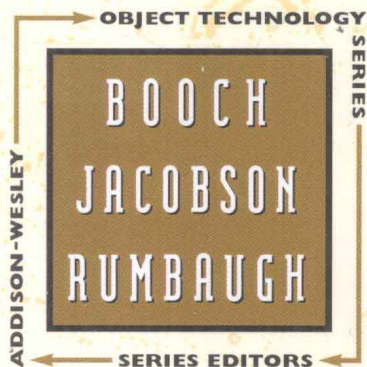
UML与面向对象设计影印丛书

X ML程序的UML建模

MODELING XML
APPLICATIONS WITH UML
PRACTICAL e-BUSINESS APPLICATIONS

(美) DAVID CARLSON 编著

Foreword by Jeffrey Hammond
Rational Software Corporation



科学出版社

www.sciencep.com



UML 与面向对象设计影印丛书

XML 程序的 UML 建模

Modeling XML Applications with UML
Practical e-Business Applications

(美) David Carlson 编著

科学出版社

北 京

图字: 01-2003-7657 号

内 容 简 介

本书通过一个大型综合应用实例,讲解如何将 XML 和 UML 结合,创建动态的 Web 应用程序,实现最优的 B2B 应用集成。全书共分 12 章,讨论了 XML 词汇表的设计与可视化分析,探讨了从 XML 词汇表生成 DTD 和 Schema 语言的过程,以及企业级集成和门户的设计方法。每章都附有一个“成功之路”,向读者提供了规划设计阶段的一些重要提示和值得注意的问题。

本书适用于 Web 系统和电子商务领域的系统分析师、事务分析师以及专业设计人员,对于 XML 和 UML 的初学者,本书也囊括了相关的概念和应用方法。

English reprint copyright©2003 by Science Press and Pearson Education Asia Limited.

Original English language title: Modeling XML Applications With UML: Practical e-Business Applications, 1st Edition by David Carlson, Copyright©2001

ISBN 0-201-70915-5

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley Publishing Company, Inc.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。

图书在版编目(CIP)数据

XML 程序的 UML 建模=Modeling XML Applications with UML:Practical e-Business Applications/(美)David Carlson 编著.一影印本.一北京:科学出版社,2004

(UML 与面向对象设计影印丛书)

ISBN 7-03-012491-X

I.X... II.C... III.①可扩充语言,XML—程序设计—英文②面向对象语言,UML—程序设计—英文 IV.TP312

中国版本图书馆 CIP 数据核字(2003)第 103043 号

策划编辑:李佩乾/责任编辑:舒立
责任印制:吕春珉/封面制作:东方人华平面设计室

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 1 月第 1 版 开本:787×960 1/16

2004 年 1 月第一次印刷 印张:22 1/4

印数:1—3 000 字数:422 000

定价:36.00 元

(如有印装质量问题,我社负责调换(环伟))

影印前言

随着计算机硬件性能的迅速提高和价格的持续下降,其应用范围也在不断扩大。交给计算机解决的问题也越来越难,越来越复杂。这就使得计算机软件变得越来越复杂和庞大。20 世纪 60 年代的软件危机使人们清醒地认识到按照工程化的方法组织软件开发的必要性。于是软件开发方法从 60 年代毫无工程性可言的手工作坊式开发,过渡到 70 年代结构化的分析设计方法、80 年代初的实体关系开发方法,直到面向对象的开发方法。

面向对象的软件开发方法是在结构化开发范型和实体关系开发范型的基础上发展而来的,它运用分类、封装、继承、消息等人类自然的思维机制,允许软件开发者处理更为复杂的问题域和其支持技术,在很大程度上缓解了软件危机。面向对象技术发端于程序设计语言,以后又向软件开发的早期阶段延伸,形成了面向对象的分析和设计。

20 世纪 80 年代末 90 年代初,先后出现了几十种面向对象的分析设计方法。其中,Booch, Coad/Yourdon、OMT 和 Jacobson 等方法得到了面向对象软件开发界的广泛认可。各种方法对许多面向对象的观念的理解不尽相同,即便概念相同,各自技术上的表示法也不同。通过 90 年代不同方法流派之间的争论,人们逐渐认识到不同的方法既有其容易解决的问题,又有其不容易解决的问题,彼此之间需要进行融合和借鉴;并且各种方法的表示也有很大的差异,不利于进一步的交流与协作。在这种情况下,统一建模语言(UML)于 90 年代中期应运而生。

UML 的产生离不开三位面向对象的方法论专家 G. Booch、J. Rumbaugh 和 I. Jacobson 的通力合作。他们从多种方法中吸收了大量有用的建模概念,使 UML 的概念和表示法在规模上超过了以往任何一种方法,并且提供了允许用户对语言做进一步扩展的机制。UML 使不同厂商开发的系统模型能够基于共同的概念,使用相同的表示法,呈现彼此一致的模型风格。1997 年 11 月 UML 被 OMG 组织正式采纳为标准的建模语言,并在随后的几年中迅速地发展为事实上的建模语言国际标准。

UML 在语法和语义的定义方面也做了大量的工作。以往各种关于面向对象方法的著作通常是以比较简单的方式定义其建模概念,而以主要篇幅给出过程指导,论述如何运用这些概念来进行开发。UML 则以一种建模语言的姿态出现,使用语言学中的一些技术来定义。尽管真正从语言学的角度看它还有许多缺陷,但它在这方面所做的努力却是以往的各种建模方法无法比拟的。

从 UML 的早期版本开始,便受到了计算机产业界的重视,OMG 的采纳和大公司的支持把它推上了实际上的工业标准的地位,使它拥有越来越多的用户。它被广泛地用

于应用领域和多种类型的系统建模,如管理信息系统、通信与控制系统、嵌入式实时系统、分布式系统、系统软件等。近几年还被运用于软件再工程、质量管理、过程管理、配置管理等方面。而且它的应用不仅仅限于计算机软件,还可用于非软件系统,例如硬件设计、业务处理流程、企业或事业单位的结构与行为建模,等等。

在 UML 陆续发布的几个版本中,逐步修正了前一个版本中的缺陷和错误。即将发布的 UML2.0 版本将是对 UML 的又一次重大的改进。将来的 UML 将向着语言家族化、可执行化、精确化等理念迈进,为软件产业的工程化提供更有力的支撑。

本丛书收录了与面向对象技术和 UML 有关的十几本书,反映了面向对象技术最新的发展趋势以及 UML 的新的研究动态。其中涉及对面向对象建模理论研究与实践的有这样几本书:《面向对象系统架构及设计》主要讨论了面向对象的基本概念、静态设计、永久对象、动态设计、设计模式以及体系结构等近几年来面向对象技术领域中的新的理论知识与方法;《用 UML 进行用况对象建模》主要介绍了面向对象的需求阶段、分析阶段、设计阶段中用况模型的建立方法与技术;《高级用况建模》介绍了在建立用况模型中需要注意的高级的问题与技术;《UML 面向对象设计基础》则侧重于经典的面向对象理论知识的阐述;《UML 参考手册》列出了 UML 的所有术语和标准元素,从语义、表示法和用途等方面详尽地介绍了 UML 的构成和概念。

涉及 UML 在特定领域的运用的有这样几本:《UML 实时系统开发》讨论了进行实时系统开发时需要对 UML 进行扩展的技术;《用 UML 构建 Web 应用程序》讨论了运用 UML 进行 Web 应用建模所应该注意的技术与方法;《面向对象系统测试:模型、视图与工具》介绍了将 UML 应用于面向对象的测试领域所应掌握的方法与工具;《对象、构件、框架与 UML 应用》讨论了如何运用 UML 对面向对象的新技术——构件-框架技术建模的方法策略;《UML 与 Visual Basic 应用程序开发》主要讨论了从 UML 模型到 Visual Basic 程序的建模与映射方法;《XML 程序的 UML 建模》讲解了如何将 XML 与 UML 结合,创建动态的 Web 应用程序,实现最优的 B2B 应用集成;《构建可扩展数据库应用程序》介绍了商务模式和数据库模式的建模方法以及集成系统的程序实现;《UML 与并行分布式实时应用程序设计》对 UML 在并行分布式实时系统开发中的应用作了全面而详细的介绍,尤其对面向对象方法解决此类系统特有的问题作了有针对性的讲解;《UML 与 J2EE 企业应用程序开发》系统介绍了使用 J2EE 开发企业级软件工程时,将 UML 建模技术应用到软件开发各个阶段的方法。

介绍面向对象编程技术的有两本书:《COM 高手心经》和《ATL 技术内幕》,深入探讨了面向对象的编程新技术——COM 和 ATL 技术的使用技巧与技术内幕。

还有一本《Executable UML 技术内幕》,这本书介绍了可执行 UML 的理念与其支持技术,使得模型的验证与模拟以及代码的自动生成成为可能,也代表着将来软件开发的一种新的模式。

总之，这套书所涉及的内容包含了对软件生命周期的全过程建模的方法与技术，同时也对近年来的热点领域建模技术、新型编程技术作了深入的介绍，有些内容已经涉及到了前沿领域。可以说，每一本都很经典。

有鉴于此，特向软件领域中不同程度的读者推荐这套书，供大家阅读、学习和研究。

北京大学计算机系 蒋严冰 博士

Foreword

While growing up I often heard the cliché, “a picture is worth a thousand words.” In our media-driven culture we are constantly exposed to a stream of images that convey different meanings. We are truly creatures dependent on visualization for survival. How then does one apply this primary skill to concepts that have no innate shape or form? Early examples of solutions to this challenge can be found in the formation of ideographs and alphabets. Throughout the centuries, the capacity to visually express the nonvisual has been a key to transforming the world we live in. As each wave of scientific transformation has progressed, a key accelerator in that transformation has been the expression of a standardized notation and a vocabulary to visually communicate intent. Examples abound in the fields of electronics, engineering, physics, and music.

Software development is a field still in its infancy. We do not yet share many of the advantages that other, more mature disciplines have. One disadvantage is that we are still developing standard notations and vocabularies for our discipline. We currently have many choices, ranging from vocabularies that describe program execution like C++ and Java, to notations that describe database design like IDEF1x, to combined notational vocabularies like the Unified Modeling Language (UML), which attempt to define the entire context of a software development exercise.

As long as multiple notations and vocabularies exist it will be necessary to “map” concepts from one language to another so that people who think in different vocabularies can communicate, and the systems that they build can interoperate. This book helps satisfy that need. Although there are many books

about UML and XML as independent vocabularies, a bridge between these two languages must be established so that XML developers can harness the visual power that the UML contains. After reading this book I think that you will agree with me that Dave does a great job of establishing a “translation guide” that, it is hoped, will serve as a primary input to eventual standardization.

So why spend time mapping UML and XML together? If one looks at current market dynamics the answer becomes clear. Over the last five years the UML has become established as the standard notation used to describe how systems are structured and how they should behave. The effect on software development has been profound. We have moved from a “Tower of Babel,” where competing factions argued over how to visually express systems design, to an industry unified by a “lingua franca.” In today’s market it would be difficult, if not impossible, to deliver an enterprise application development tool that did not include UML-based visualization. The result has been an explosion of productivity as different groups and companies have mapped the UML into specific domains. Examples include UML profiles for Real-time systems, Data Modeling, Enterprise JavaBeans, and just about every modern implementation language. These mappings raise the level of abstraction that today’s software development team works at.

Although the UML has become established as the standard notation for enterprise development, the very nature of enterprise development is changing in order to embrace the World Wide Web. The Web has changed the way systems are designed, and many new languages are spoken there. HTML and CGI led the first wave of Web applications, but the next wave is likely to be dominated by architectures based on the Java 2 Enterprise Edition and VS.Net. Because XML figures prominently in both architectural approaches, regardless of which path developers take, they will use XML to build solutions. A translation manual between the worlds of UML and XML is very helpful because it will accelerate the efforts of software developers who want to create Enterprise Web Applications.

Accelerating the efforts of today’s XML developer is key because the demands they face are greater than ever. Brick and mortar companies are struggling to redefine themselves as “click and mortar” concerns and view software as critical to this effort. This sets up a “software paradox” where XML developers must deliver solutions more quickly than in the past (hence the term “internet time”) and yet these solutions must be high quality because the customer interfaces with them directly. The fact that many business-to-customer and business-to-business solutions use XML as a communication language makes it imperative to accelerate the efforts of XML developers. By automating the process by which visual diagrams (in UML) become XML schemas and DTDs, we can accelerate e-business development while retaining a high level of quality in the resulting schemas.

Ironically, when I reviewed the draft of this publication our team at Rational had just delivered XML DTD support for Rational Rose. Any developer who has built a language generator for a UML tool (our team has built four to date) will tell you that the mapping is usually the critical part of this effort. Once a mapping is established the work goes pretty quickly. Because we had engaged in our own mapping exercise I was quite eager to see how our work stacked up. Although there were some small differences (for example, use of N-ary associations, and treatment of Entities), I was relieved to find that we had made most of the same connections that Dave has. If we'd had this text before we started, we would have had a great cheat sheet for our work. In particular, we found Dave's treatment of XPath, XPointer and XLink very useful for our continued work in evolving our own UML-XML support into the future. Based on our work in both the UML and XML community, our team feels that this book represents the best translation guide available for those looking to tightly bind the powerful expressiveness of UML and the ubiquity of XML together in a way that will help accelerate the delivery of enterprise Web applications. Enjoy!

*Jeffrey Hammond,
Senior Product Manager,
Rational Software*

Preface

Writing about XML and e-Business is a lot like taking a snapshot of a speeding train. And for those readers who are new to one or both of these subjects, it's a lot like attempting to jump onto that train. In writing this book, I've attempted to strike a balance between an introduction to these challenging subjects and a practical guide for designing realistic systems.

I make some assumptions about a basic prior knowledge of both XML and UML, but not so much that a motivated reader cannot easily meet these expectations with quick supplemental study. There are dozens of introductory books on both subjects but there is a lack of good explanation about how XML and UML can be combined in the analysis of complex systems. The goal of this book is not only to teach you *about* XML and UML but also how to *use* these technologies for practical applications.

Goals of This Book

Over the past twenty-five years of learning, teaching, and working, I have realized that there is a very significant difference between gaining knowledge about a subject and gaining *actionable* knowledge about that same topic. Knowledge is actionable when it directly and immediately affects what you do and how you do it. While writing this book, I had a note taped to the top of my computer monitor that read "Actionable Knowledge," so that it would continually prompt me to keep this focus in mind.

After reading this book, you will have learned the following actionable knowledge:

- Guidelines that you can use to gather key stakeholder input while developing your XML application.
- How you can integrate XML and UML in *current* design projects and what this means to achieving your e-business objectives.
- Steps and criteria to use in the visual analysis and design of XML vocabularies.
- A detailed guide to how you can generate XML DTDs and Schemas from those vocabularies, plus the trade-offs you must consider while doing so.
- Substantial, realistic examples to base your own work on.
- Concrete suggestions about how to apply recently adopted (or almost adopted) XML standards.
- A deep understanding that is based not on the marketing materials of individual vendors but on common practice that applies to all of them.
- A solid grounding about how to design XML applications now and many product or system releases in the future.
- An understanding of what's going to happen next!

Concepts of UML modeling and a streamlined Unified Process are woven throughout this book. e-Business examples demonstrate the breadth of UML modeling capabilities but without overwhelming the primary goal of creating successful applications using XML. As a means to this goal, this book focuses on a consistent, substantial example about the analysis and design of a product catalog application. An XML vocabulary for the Catalog Markup Language (CatML) is designed first in UML, then generated to both DTD and XML Schema languages.

This same catalog example is used to model requirements for the “MyCat” Web portal application, whose content is defined by the CatML vocabulary. An example MyCat portal is demonstrated using the Extensible Stylesheet Language Transformation (XSLT) to produce an HTML presentation from the XML documents, all based on the CatML vocabulary definitions. UML is used throughout the exercise to analyze the application requirements and the vocabulary design. Finally, XSLT is described as a language for transforming the CatML vocabulary to and from RosettaNet product catalog standards. Vocabulary transformation is an essential element in most e-business applications.

Who Should Read This Book?

This book is not a guide to programming XML applications; rather it focuses on the thoughtful analysis and design of XML vocabularies and their use within distributed systems. If you have a need to develop a system using XML, or if you are considering the value of such a system, then you will find this book helpful. Although their use is not restricted to e-business applications, those examples form the central theme throughout all chapters. These examples span the range of XML applied to the content of portal presentations to the specification and transformation of message content for system integration.

System architects will find many valuable points to consider when planning the use of XML. The use case analyses in Chapters 1, 4, and 5 build a business case for e-business integration and portal design using XML. These use cases are described from the perspective of key stakeholders who determine and evaluate the goals of a successful XML application. Each chapter concludes with a list of “Steps for Success” that are especially valuable to an architect.

Complex XML vocabulary definitions are often easier to comprehend and discuss with others when they are expressed graphically. Although a few existing tools provide some assistance in this regard, they are generally limited to a strict hierarchical view of the vocabulary structure. Complex structures may be represented in schemas that are more easily analyzed from an object-oriented perspective. These object-oriented models of schema definition are easily represented using UML class diagrams. This book is valuable to business analysts, who are responsible for the definition of business vocabularies that will be implemented using XML.

Those analysts often team with designers who fine-tune the vocabularies for generation to XML DTDs or Schemas. Chapter 8 provides a detailed comparison of XML DTDs with the new, much richer possibilities available in XML Schema definitions. Chapter 9 includes detailed design heuristics for generating both DTDs and Schemas from UML class models and describes trade-offs for specifying relaxed versus strict schema validation. These decisions are the daily work of XML designers.

Chapter 2 provides an overview of XML terminology using a simple real-world example that is relevant to the topics of this book. The Rich Site Summary (RSS) is described and compared with similar use of news content in HTML. For a more thorough introduction to XML, I recommend:

- Simon St. Laurent. *XML Elements of Style*. New York: McGraw-Hill, 1999.

If you are not familiar with UML, Chapter 3 includes a fast-paced overview of the essential diagrams that are used in this book. Those diagrams are applied to the same RSS XML example that is introduced in Chapter 2. For a good introduction to UML that is short and easy to read, I recommend:

- Martin Fowler, Kendall Scott. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Second Edition. Boston: Addison-Wesley, 2000.

Chapters 10 and 11 include substantial examples of XSLT vocabulary transformations. XSLT is a very powerful but somewhat complex language whose detail is beyond the scope of this book. If you are new to XSLT, I recommend the following supplemental references:

- Neil Bradley. *The XSL Companion*. Boston: Addison-Wesley, 2000.
- Michael Kay. *XSLT Programmer's Reference*. Birmingham, UK: Wrox Press, 2000.

XMLModeling.com

Because no book covering the topics of XML and e-business can promise more than a snapshot of the speeding train, it's equally important to offer a first-class ticket for the ride into the future. A Web site has been especially prepared as the companion site for this book. It is available at <http://XMLModeling.com>.

The following information is available, organized in an easily navigated portal:

- Current XML news
- Quick links for XML and UML resources and tools
- Complete UML models and XML listings from this book's examples
- Tips and tools for generating XML schemas from UML models
- Case studies that apply these techniques

Acknowledgments

I am very grateful for the support of all my friends and colleagues at Gartner Solista in Boulder. I especially wish to thank Maryann Richards for providing detailed feedback on every chapter within days of its first draft. Her suggestions have made positive contributions to every chapter of this book. Also, Tim Johnson read the first versions of each chapter and helped keep me focused on the planned audience.

A special thanks to Kristin Erickson from Addison-Wesley for convincing me to write this book, then coaching me through the publication process and orchestrating a team of very busy reviewers. And of course I'm very grateful for the time and input of the anonymous reviewers who made significant contributions to these chapters.

And last, but certainly not least, the Bookend Café and the Trident coffee-house in Boulder, Colorado, deserve honorable mentions. A substantial part of this book was imagined, written, or edited in those cafes while drinking many gallons of tea and eating more than a hundred chocolate-chip cookies.

David Carlson, Ph.D.
Boulder, Colorado

Contents

Foreword *xi*

Preface *xv*

PART I FOUNDATIONS 1

Chapter 1	Convergence of Communities	3
	Models for e-Business	4
	Stakeholder Communities	5
	Consumer	6
	Business Analyst	8
	Web Application Specialist	9
	System Integration Specialist	10
	Content Developer	10
	Road Map for This Book	11
	Part I. Foundations	11
	Part II. XML Vocabularies	11
	Part III. Deployment	12
	Steps for Success	13
Chapter 2	What Is an XML Application?	15
	HTML, XML, and XHTML	15
	XML Vocabularies	18

	XML Presentation	23
	Cascading Style Sheets	23
	XSLT Stylesheets	26
	Chapter Summary	29
	Steps for Success	30
Chapter 3	What Is a UML Model?	31
	Models and Views	32
	Requirements Workflow	33
	Use Case Diagram	33
	Analysis Workflow	35
	Activity Diagram	35
	Model Management Diagram	37
	Collaboration Diagram	37
	Design Workflow	39
	Class Diagram	40
	Object Diagram	43
	Sequence Diagram	45
	Component Diagram	46
	The Unified Process	47
	Chapter Summary	49
	Steps for Success	50
Chapter 4	e-Business Integration with XML	51
	Use Case Analysis	52
	Catalog Vocabulary Requirements	54
	Shared Business Vocabularies	55
	Define Business Vocabulary	55
	Create XML Schema	56
	Validate Message	59
	Transform Message Content	60
	Process Workflow and Messaging	61
	Define Business Process	61
	Build Workflow Model	62
	Define Message Protocol	64
	Application Integration	65
	Create Application Classes	65
	Create Legacy Adapter	67
	Chapter Summary	68
	Steps for Success	69
Chapter 5	Building Portals with XML	71
	Use Case Analysis	72

Content Management	74
Define Business Vocabulary	74
Create Content	74
Assign Content Metadata	77
Portal Design	77
Design Portlet	78
Design Content Template	80
Create Stylesheet	81
Design Portal Layout	83
Customize Portal Layout	83
Wired and Wireless Convergence	84
Chapter Summary	86
Steps for Success	87

PART II XML VOCABULARIES 89

Chapter 6	Modeling XML Vocabularies	91
	What Is a Vocabulary?	91
	CatML Vocabulary	93
	Simplified Product Catalog Model	94
	Mapping UML to XML	95
	XML Metadata Interchange	97
	Disassembling UML Objects into XML	99
	UML Classes to XML Elements	99
	Inheritance	101
	UML Attributes to XML Elements	102
	UML Attributes to XML Attributes	103
	Enumerated Attribute Values	106
	Mapping UML Compositions	107
	Mapping UML Associations	109
	Roots and Broken Branches	111
	Packaging Vocabularies	113
	FpML Vocabulary	114
	UML Packages	115
	XML Namespaces	117
	Chapter Summary	119
	Steps for Success	120
Chapter 7	From Relationships to Hyperlinks	121
	Expanded CatML Vocabulary	122
	XML Standards for Linking	125
	XML ID and IDREF	126
	XPath	128