

CARL HANSER VERLAG

Horst Neumann

**Analyse und Entwurf von Softwaresystemen mit der
UML**

3-446-22038-0

www.hanser.de

Vorwort

Anwendung der Unified Modeling Language (UML) heißt Entwicklung von Software-Systemen unter Ausschöpfung der objektorientierten Konzepte und Verfahren, die ein mächtiges Potenzial für das Software Engineering bieten. Während sich diverse Programmiersprachen für die *Implementierung* von Entwurfseinheiten anbieten, bildet die UML die gemeinsame Grundlage für werkzeuggestützte Methoden, welche die *Analyse der Systemnutzung*, die *objektorientierte Anforderungsanalyse*, den *objektorientierten Entwurf*, die *Spezifikation der Entwurfseinheiten*, die *automatische Codegenerierung* und den *systematischen Test* wirkungsvoll unterstützen.

Das vorliegende Buch ist eine völlig überarbeitete Fassung des Buchs „Objektorientierte Softwareentwicklung mit der Unified Modeling Language (UML)“. Die Struktur der Darstellung und die Behandlung der diversen Modellelemente und Modelle wurden an die typische Anwendung der UML in einem Projekt zur Entwicklung eines Software-Systems angepasst, sodass die Verständlichkeit und der Nutzeffekt des Buchs sich deutlich erhöhen. Außerdem sind zahlreiche Erfahrungen aus der Schulung und praktischen Anwendung der UML eingeflossen und viele Hinweise von Lesern für Korrekturen und Vorschläge für eine verbesserte Darstellung berücksichtigt.

Die Nutzung der UML wird durch die Object Management Group (OMG) gefördert. Der Prozess der Standardisierung im Hinblick auf die praktische Anwendung der UML ist weitgehend abgeschlossen. Die Grundlage für das vorliegende Buch bildet das im September 2001 publizierte Dokument „OMG Unified Modeling Language Specification – Version 1.4“, in der alle für eine umfassende und solide Anwendung relevanten Konstrukte in einer konsistenten Form definiert sind. Die UML wird von zahlreichen Anwendern bereits in Projekten mit großem Erfolg genutzt. Die führenden Werkzeughersteller bieten moderne Werkzeuge zur wirkungsvollen Unterstützung der Entwicklung. Man kann mit Recht behaupten, dass die UML das Software Engineering maßgeblich beeinflusst und die Voraussetzungen für eine ingenieurgemäße Disziplin geschaffen hat.

Die volle Nutzung der UML für die Entwicklung von Software-Systemen setzt voraus, dass man die zugrunde liegenden Konzepte, die Syntax und Semantik der UML-Modellelemente/-Modelle und die Vorgehensweise bei der Anforderungsanalyse, beim Entwurf und bei der Realisierung des Entwurfs verstanden hat. Einen wesentlichen Beitrag zu diesem Verständnis zu leisten, ist das Anliegen des vorliegenden Buches. Das Buch ist deshalb linear konzipiert, d. h. die einzelnen Kapitel bauen aufeinander auf. Die Nutzung des Buchs als Referenzdokument wird durch zahlreiche Vorwärts- und Rückwärtsreferenzen unterstützt.

Das Buch richtet sich an alle, die eine sachliche, systematische, umfassende und verständliche Darstellung der objektorientierten Entwicklung von Software-Systemen mit

der UML suchen, weil sie das mächtige Potenzial besser verstehen und sich für die eigene Nutzung erarbeiten wollen. Dies trifft insbesondere zu auf:

- Studenten der Informatik an Universitäten und Fachschulen;
- Studenten der Mathematik, der Physik und der Ingenieurwissenschaften, für die solide Kenntnisse im Bereich der Software-Entwicklung erforderlich bzw. nützlich sind;
- Studierende mit Schwerpunkt Datenverarbeitung an Wirtschaftsfachschulen;
- Fachleute in der Industrie: Systemanalytiker, Systemarchitekten, Entwickler von Software-Systemen, Projektmanager und Beauftragte der Qualitätssicherung.

Der Leser sollte über gewisse Erfahrungen in der Software-Entwicklung mit einer höheren Programmiersprache und über Grundkenntnisse des Software Engineering verfügen. Nützlich, aber nicht Voraussetzung ist die Kenntnis einer objektorientierten Programmiersprache, z. B. C++, Java, Smalltalk, Ada.

Insgesamt werden im Buch zur Erläuterung der objektorientierten Entwicklung mit der UML zahlreiche Beispiele und über 210 UML-basierte Abbildungen verwendet. Dies zeigt die Bedeutung einer verständlichen grafischen Repräsentation. Da nicht nur die Grundlagen, sondern auch wesentliche Zusammenhänge und die praktische Anwendung der UML vermittelt werden, wird der Leser tatsächlich in die Lage versetzt, mit hoher Produktivität qualitativ hochwertige Software zu erzeugen.

Die objektorientierte Terminologie ist uneinheitlich, was auch für den Begriff *objektorientiert* gilt. Die Basis-Begriffe, die neben dem zentralen Begriff *Objekt* eine wesentliche Rolle spielen, sind *Klasse* und *Typ*, die diverse Abstraktionen bezeichnen, sowie *Vererbung*, *Assoziation*, *Instanz*, *Botschaft* usw. Um den Leser eine Hilfe zu bieten, sind die Definitionen der wesentlichen Begriffe in einem Glossar im Anhang des Buches zusammengefasst.

Im Buch werden Abkürzungen verwendet, die im Kontext des objektorientierten Ansatzes gebräuchlich sind, wie z. B. OOA, OOE, OOPS. Die Abkürzungen sind zusammen mit dem voll ausgeschriebenen Text am Ende des Glossars aufgelistet.

Die Fachbücher, Fachartikel und die UML-Spezifikationen, die durch relevante Informationen zu diesem Buch beigetragen haben oder für dieses Buch zu Rate gezogen wurden, sind im Literaturverzeichnis zusammengestellt. Das Verzeichnis ist nach Sachgebieten entsprechend den Kapiteln des Buchs und ansonsten alphabetisch nach Autoren geordnet.

Zu diesem Buch haben zahlreiche Personen direkt oder indirekt beigetragen. An erster Stelle möchte ich mich bei meiner Lebensgefährtin, Frau Marianne Buchholz, bedanken, die mir bei der Erstellung und Korrektur des Manuskripts geholfen hat. Zur Beseitigung von Wissenslücken und zur Klärung von Fragen haben Diskussionen mit Fachleuten und Teilnehmern an relevanten Schulungskursen beigetragen. Mein besonderer Dank gilt Frau Margarete Metzger vom Carl Hanser Verlag, die mein Anliegen unterstützt hat, ein neues Buch zu machen, welches das Thema umfassend behandelt.

Einführung

In the same way a blueprint helps a team collaborate successfully on constructing a building, the Unified Modeling Language helps a team visualize an application's architecture throughout the development lifecycle.

Rational Software Corporation
(UML Resource Center)

Die *objektorientierten* Konzepte und die darauf basierenden Verfahren haben dazu beigetragen, dass die mangelnde Unterstützung der Anforderungsanalyse und des Entwurfs und andere Einschränkungen der herkömmlichen Entwicklungsmethoden und Programmiersprachen beseitigt wurden. Die *Objekttechnologie* ermöglicht deshalb die Entwicklung und Wartung von Software-Systemen im Sinne einer ingenieurgemäßen Disziplin, da die wesentlichen Grundlagen für ein solides *Software Engineering* geschaffen wurden. Die Folge ist, dass die Objekttechnologie heute den gemeinsamen Nenner bildet für die Vielfalt der Software-Systeme, die in den verschiedenen Anwendungsbereichen entstehen. Besonders deutlich zeigt sich dies bei den typischen *Subsystemen*, zu denen vor allem die grafischen Benutzeroberflächen, Datenbanksysteme, wissensbasierte Systeme und vernetzte Kommunikationssysteme gehören, die trotz ihrer unterschiedlichen Anforderungen auf gemeinsame objektorientierte Konzepte zurückgeführt werden.

Die objektorientierten Konzepte bilden auch die Grundlage der *Unified Modeling Language* – kurz *UML* –, die als Standard einer grafischen Notation zur Modellbildung konzipiert ist. Die wesentlichen Bestandteile der UML sind diverse Modellelemente mit zugeordneten grafischen Symbolen, deren Semantik weitgehend formal und so präzise wie möglich definiert ist. Die UML zeichnet sich außerdem durch ein neuartiges Konzept aus, das Möglichkeiten für Erweiterungen bietet. Aus vordefinierten Modellelementen können Modellelemente mit spezialisierter Semantik abgeleitet und damit Modelle für spezielle Zwecke gebildet werden.

Die UML unterstützt wirkungsvoll die Konzipierung und Darstellung von Analyse- und Entwurfsergebnissen in Form von intuitiv erfassbaren Modellen und Spezifikationen, die sich besonders gut zur Analyse der Anforderungen und zum Nachweis ihrer Berücksichtigung eignen. Gleiches gilt für den Entwurf zur Gestaltung einer flexiblen Aufbau- und Ablaufstruktur, die bei umfangreichen und komplexen Software-Systemen die korrekte und zuverlässige Wechselwirkung der zahlreichen Komponenten sicherstellt, selbst wenn diese auf lokal oder regional vernetzte Datenverarbeitungsanlagen verteilt sind. Die UML bildet damit das einheitliche und eindeutige Mittel der Verständigung von Systemexperten, Anwendern und Entwicklern. Ihre Bedeutung

ist deshalb höher einzuschätzen als die einer objektorientierten Programmiersprache, die zur rechnergerechten Codierung des Entwurfs eingesetzt wird. Die deklarativen Konstrukte und wesentliche Teile des ablauffähigen Codes können nämlich direkt von einem Codegenerator aus den UML-Modellen und -Spezifikationen abgeleitet werden.

Die UML ist unabhängig von einer bestimmten Methode, unabhängig von einer Programmiersprache und unabhängig von einem speziellen Entwicklungsprozess konzipiert. Eine Untermenge der UML-Konstrukte ist auf eine objektorientierte Programmiersprache abbildbar. Der angemessene Entwicklungsprozess wird getrieben durch den Einsatz fortschrittlicher Methoden, durch definierte Nutzungsfälle und durch eine flexible Systemarchitektur, ist inkrementell, iterativ und rekursiv.

Die objektorientierten Schlüsselkonzepte, welche die Grundlage der Objekttechnologie und auch der UML bilden, werden im ersten Kapitel dargestellt.

Gegenstand von Kapitel 2 ist die Umsetzung des objektorientierten Ansatzes in einen praxisgerechten Entwicklungsprozess, der die Voraussetzungen schafft für die iterative bzw. evolutionäre Entwicklung von Software-Systemen mit hoher Qualität, eine hohe Leistungsfähigkeit der Entwickler garantiert und eine verlässliche Planung und Kontrolle ermöglicht.

Kapitel 3 vermittelt einen Überblick über die objektorientierten Methoden. Jede Methode definiert eine spezielle Vorgehensweise für die schrittweise Konstruktion von Modellen, welche die Anforderungsanalyse und/oder den Entwurf des Software-Systems unterstützen. Einige Methoden, die eine weite Verbreitung gefunden und wesentlich zur Konzipierung der UML beigetragen haben, werden näher betrachtet.

Gegenstand von Kapitel 4 sind die vordefinierten Modellelemente der UML, das UML-Metamodell und die Object Constraint Language (OCL). Ein wesentlicher Aspekt sind die Möglichkeiten zur Bildung von Modellelementen mit spezialisierter Semantik. Den Abschluss bildet ein Überblick über die diversen UML-Modelle und die zugeordneten Diagramme, die für die Analyse, den Entwurf und die Realisierung eines Software-Systems genutzt werden.

In Kapitel 5 wird die Analyse des Systemverhaltens behandelt. Eine wesentliche Methode ist die Analyse der Nutzungsfälle (Use-Case-Analyse). Die Vorgehensweise, die sich im praktischen Einsatz bewährt hat, wird umfassend erläutert. Ein weiteres Thema ist der Entwurf der Systemarchitektur.

Kapitel 6 beschreibt die Durchführung der objektorientierten Analyse unter Nutzung der UML für die Konzipierung der diversen Analysemodelle. Dabei werden die für die Analyse charakteristischen Iterationen im Entwicklungsprozess umfassend erläutert und dafür typische Beispiele verwendet.

Schwerpunkt von Kapitel 7 ist der objektorientierte Entwurf. Zunächst werden die objektorientierten Entwurfsprinzipien und Entwurfstechniken behandelt. Danach wird gezeigt, wie die Analysemodelle zu einer flexiblen und realisierbaren Aufbau- und Ab-

laufstruktur erweitert werden. Dabei werden die für den Entwurf charakteristischen Iterationen im Entwicklungsprozess umfassend erläutert.

Kapitel 8 demonstriert die Nutzung der UML für die Analyse und den Entwurf anhand zweier Fallbeispiele. Das erste ist eine betriebswirtschaftliche Anwendung mit interaktiver Systemnutzung, das zweite eine technische Anwendung mit parallelen Abläufen unter strengen Zeitbedingungen.

Kapitel 9 vermittelt einen Überblick über die wichtigsten Aspekte der Realisierung eines Systems auf der Grundlage des Entwurfs.

Anhang A1 bietet eine Übersicht über die wichtigsten Werkzeuge, welche die Anforderungsanalyse und den objektorientierten Entwurf von Software-Systemen auf der Grundlage der UML unterstützen. In einer Tabelle sind Werkzeuge und Anbieter (entsprechend dem Stand von 11/2001) angeführt. Demo-Versionen und Beispiel-Modelle zur Erprobung eines Werkzeugs sind vom jeweiligen Anbieter oder direkt über das Internet verfügbar.

Anhang A2 enthält ein umfangreiches Glossar mit allen wichtigen Begriffen, Anhang A3 das Abkürzungsverzeichnis. In Anhang A4 finden Sie ein umfassendes Verzeichnis mit relevanter Literatur.

Die OMG-Spezifikation der UML – Version 1.4 – sowie Informationen der UML Revision Task Force (RTF) können über das Internet abgerufen werden, und zwar über die Adressen <http://www.omg.org> bzw. <http://www.celigent.com/omg/umlrtf>. Der Autor nimmt Fragen zum Buch, Hinweise und Verbesserungsvorschläge per Email entgegen: horst.neumann@t-online.de.