

Grady Booch  
James Rumbaugh  
Ivar Jacobson

# Das UML Benutzerhandbuch

Aktuell zur Version 2.0

# 7 Diagramme

## In diesem Kapitel:

- ▶ Diagramme, Sichten und Modelle
- ▶ Verschiedene Sichten eines Systems modellieren
- ▶ Verschiedene Abstraktionsebenen modellieren
- ▶ Komplexe Sichten modellieren
- ▶ Diagramme und weitere Artefakte modellieren

Beim Modellieren erstellen Sie ein vereinfachtes Abbild der Realität mit der Absicht, das zu entwickelnde System besser zu verstehen. Mit UML erstellen Sie Ihre Modelle aus einfachen Grundbausteinen wie Klassen, Interfaces, Kollaborationen, Komponenten, Knoten, Abhängigkeiten, Generalisierungen und Assoziationen.

*Modellierung  
wird in Kapitel 1  
behandelt.*

Diagramme sind die Hilfsmittel, die Ihnen eine Betrachtung dieser Bausteine ermöglichen. Ein Diagramm ist eine grafische Abbildung einer Reihe von Elementen, die am häufigsten als verbundener Graph aus Knoten (Dingen) und Bögen (Beziehungen) dargestellt werden. Diagramme werden eingesetzt, um das System aus verschiedenen Perspektiven zu visualisieren. Da ein komplexes System nicht über eine einzige Perspektive vollständig verstanden werden kann, sind in UML eine Anzahl von Diagrammen definiert, die es Ihnen ermöglichen, sich unabhängig voneinander auf unterschiedliche Aspekte Ihres Diagramms zu konzentrieren.

Gute Diagramme machen das System, das Sie entwickeln, verständlich und übersichtlich. Die Auswahl der richtigen Diagrammarten zur Modellierung des Systems zwingt Sie dazu, die richtigen Fragen zu Ihrem System zu stellen, und hilft Ihnen dabei, die Auswirkungen Ihrer Entscheidungen abzuwägen.

## 7.1 Erste Schritte

Wenn Sie mit einem Architekten an der Entwicklung eines Hauses arbeiten, gehen Sie von drei Dingen aus: einer Wunschliste (beispielsweise »Ich möchte ein Haus mit drei großen Zimmern« und »Ich möchte höchstens x bezahlen«), einigen einfachen Zeichnungen oder Bildern, die einige der Schlüsselfunktionen darstellen (beispielsweise eine Abbildung eines Eingangs mit einer Wendeltreppe) und einer groben Vorstellung vom Stil des Hauses (beispielsweise »Wir hätten gern den Französischen Landhausstil mit einem Hauch Kalifornische Küste«). Aufgabe des Architekten ist es, diese unvollständigen, veränderlichen und vielleicht sogar gegensätzlichen Anforderungen in einem Design zu vereinen.

Um dies zu erreichen, geht der Architekt wahrscheinlich von einer Blaupause des Grundrisses aus. Dieses Artefakt bietet Ihnen und Ihrem Architekten die Möglichkeit, sich das fertige Haus vorzustellen, Einzelheiten festzulegen und getroffene Entscheidungen aufzuzeichnen. Bei jeder Überarbeitung werden Sie Änderungen vornehmen wollen, beispielsweise Wände verschieben, Räume neu anordnen und Fenster und Türen anordnen. Anfangs ändern sich diese Blaupausen noch häufig. Während die Entwicklung voranschreitet und Sie zuversichtlich sind, dass Sie nun über ein Design verfügen, das den Anforderungen von Form, Funktion, Zeit und Geld entspricht, stabilisieren diese Blaupausen sich bis zu einem Punkt, an dem mit dem Bau des Hauses begonnen werden kann. Auch während Ihr Haus gebaut wird, ändern Sie wahrscheinlich einige der Diagramme oder erzeugen sogar einige neue.

Während des Vorgangs möchten Sie andere Ansichten des Hauses sehen als den einfachen Grundriss, beispielsweise Gesamtaußenansichten, auf denen die Seiten des Hauses gezeigt sind. Wenn Sie mit dem Festlegen der Details beginnen, muss Ihr Architekt, um einen aussagekräftigen Kostenvoranschlag erstellen zu können, elektrische Pläne, Pläne für Heizung und Belüftung sowie Pläne für die Frischwasserversorgung und Abwasseranschlüsse erstellen. Erfordert Ihre Planung einige besondere Konstruktionsmerkmale (beispielsweise einen langen, selbsttragenden Bereich über dem Keller) oder ist eine Funktion für Sie besonders wichtig (beispielsweise eine besondere Anordnung eines Kamins, damit Sie ein Heimkino einrichten können), werden Sie zusammen mit Ihrem Architekten einige Zeichnungen erzeugen wollen, die diese Einzelheiten hervorheben.

Die Vorgehensweise, Systeme mithilfe von Diagrammen aus unterschiedlichen Perspektiven darzustellen, ist nicht auf die Bauindustrie beschränkt. Sie wird Ihnen in allen Ingenieurwissenschaften begegnen, zu denen die Erstellung komplexer Systeme gehört, von Städtebau zum Flugzeugbau, Schiffbau, der Produktionsindustrie und der Softwareentwicklung.

Im Zusammenhang der Software gibt es fünf sich ergänzende Sichten, die beim Visualisieren, Spezifizieren, Konstruieren und Dokumentieren eines Softwaresystems von besonders großer Bedeutung sind: die Anwendungsfallsicht, die Entwurfssicht, die Interaktionssicht, die Implementierungssicht und die Verteilungssicht. Jede dieser Sichten umfasst sowohl Strukturmodellierung (die Modellierung statischer Dinge) als auch Verhaltensmodellierung (die Modellierung dynamischer Dinge). Zusammen erfassen diese Sichten die wichtigsten Entscheidungen über ein System. Einzelne bieten diese Sichten Ihnen die Möglichkeit, sich auf eine Betrachtungsperspektive des Systems zu konzentrieren und sich so Klarheit über die Entscheidungen zu verschaffen.

*Die fünf Sichten einer Architektur werden in Kapitel 2 behandelt.*

Beim Betrachten eines Softwaresystems aus einer beliebigen Perspektive mit UML setzen Sie Diagramme ein, um die interessanten Elemente zu organisieren. UML definiert verschiedene Diagrammarten, die Sie zum Aufbau der Sichten nach Bedarf zusammenfügen und aufeinander abstimmen können. Die statischen Aspekte der Implementierungssicht eines Systems beispielsweise lassen sich möglicherweise über Klassendiagramme visualisieren, während sich die dynamischen Aspekte derselben Implementierungssicht möglicherweise über Interaktionsdiagramme visualisieren lassen.

*Die Modellierung einer Systemarchitektur wird in Kapitel 32 behandelt.*

Natürlich müssen Sie sich nicht auf die vordefinierten Diagrammtypen beschränken. In UML sind einige Diagrammtypen vorbereitet, weil es sich bei ihnen um die häufigsten Pakete von betrachteten Elementen handelt. Um den Anforderungen Ihres Projekts oder Ihrer Organisation gerecht zu werden, können Sie Ihre eigenen Diagrammtypen erzeugen, um UML-Elemente auf verschiedene Weise zu betrachten.

Sie werden die Diagramme von UML auf zwei grundlegende Arten anwenden: um Modelle zu spezifizieren, von denen aus Sie ein ausführbares System erstellen (Forward-Engineering), und um Teile eines ausführbaren Systems in Modelle zurückzuführen (Reverse-Engineering). Von welcher Richtung Sie auch ausgehen, Sie werden doch, genau wie ein Bauarchitekt, dazu neigen, Ihre Diagramme inkrementell (Stück für Stück) und iterativ (den Vorgang »ein wenig aufbauen, ein wenig planen« wiederholend durchführen) zu erstellen.

*Dieser inkrementelle und iterative Vorgang wird in Anhang B zusammengefasst.*

### 7.1.1 Grundgedanken und Begriffe

Ein *System* ist eine Anordnung von Subsystemen zur Erfüllung eines Zweckes und lässt sich durch eine Reihe von Modellen beschreiben, die von unterschiedlichen Gesichtspunkten ausgehen können. Ein *Subsystem* ist eine Anordnung von Elementen, von denen einige eine Spezifikation des Verhaltens bilden, das die anderen enthaltenen Elemente bieten. Ein *Modell* ist eine semantisch abge-

*Systeme, Modelle und Sichten werden in Kapitel 32 behandelt.*

schlossene Abstraktion eines Systems, was bedeutet, dass es sich um eine vollständige und zusammenhängende vereinfachte Darstellung der Realität handelt, die für ein besseres Verständnis der Realität erzeugt wurde. Im Kontext der Architektur ist eine *Sicht* die Darstellung von Aufbau und Organisation eines Modells, die auf einen Aspekt des Systems konzentriert ist. Ein *Diagramm* ist eine grafische Abbildung einer Menge von Elementen, die am häufigsten als verbundener Graph aus Knoten (Dingen) und Bögen (Beziehungen) dargestellt wird.

Mit anderen Worten, ein System stellt das zu entwickelnde Ding aus verschiedenen Perspektiven verschiedener Modelle dar, deren Sichten in Form von Diagrammen dargestellt werden.

Ein Diagramm ist nichts weiter als die grafische Darstellung der Elemente, aus denen ein System besteht. Beispielsweise besitzt das System für die Personalabteilung eines großen Unternehmens möglicherweise mehrere hundert Klassen. Struktur und Verhalten dieses Systems kann man sich nicht vorstellen, indem man auf ein großes Diagramm starrt, das all diese Klassen und ihre Beziehungen enthält. Stattdessen empfiehlt es sich, verschiedene Diagramme zu erzeugen, die jeweils auf eine Sicht(-weise) konzentriert sind. Beispielsweise finden Sie ein Klassendiagramm mit den Klassen `Person`, `Abteilung` und `Büro` vor, das zur Konstruktion eines Datenbankschemas zusammengestellt worden ist. Einige dieser Klassen begegnen Ihnen dann, zusammen mit einigen weiteren, erneut in einem Diagramm von einer Programmierschnittstelle, auf die Client-Anwendungen zugreifen. Höchstwahrscheinlich finden Sie dann einige dieser Klassen auch in einem Interaktionsdiagramm wieder, das die Semantik einer Transaktion abbildet, über die eine Person einer neuen Abteilung zugewiesen wird.

Wie dieses Beispiel zeigt, kann das gleiche Ding in einem System (beispielsweise die Klasse `Person`) mehrfach im gleichen oder in unterschiedlichen Diagrammen erscheinen. In jedem Fall handelt es sich um dasselbe Ding. Jedes Diagramm bietet eine Sicht der Elemente, aus denen ein System aufgebaut ist.

Bei der Modellierung realer Systeme werden Sie unabhängig vom Problembereich feststellen, dass Sie immer wieder die gleichen Diagrammartentypen erstellen, da diese allgemeine (An-)sichten von allgemeinen Modellen bieten. Üblicherweise werden Sie die statischen Teile eines Systems mithilfe eines der folgenden Diagramme betrachten:

1. Klassendiagramm
2. Komponentendiagramm
3. Montagediagramm
4. Objektdiagramm

5. Verteilungsdiagramm
6. Artefaktendiagramm

Zur Betrachtung der dynamischen Teile eines Systems werden Sie häufig fünf weitere Diagramme einsetzen:

1. Anwendungsfalldiagramm
2. Sequenzdiagramm
3. Kommunikationsdiagramm
4. Zustandsdiagramm
5. Aktivitätsdiagramm

Jedes von Ihnen erstellte Diagramm wird wahrscheinlich eines dieser neun sein oder gelegentlich eines einer anderen Art, die genau für Ihr Projekt oder Ihre Organisation definiert ist. Jedes Diagramm muss einen Namen besitzen, der im Kontext eindeutig ist und mit dessen Hilfe Sie auf ein bestimmtes Diagramm zugreifen und die Diagramme voneinander unterscheiden können. In allen Systemen, die über die einfachsten hinausgehen, empfiehlt es sich, die Diagramme in Pakete zu organisieren.

*Pakete werden  
in Kapitel 12  
behandelt.*

Sie können eine beliebige Kombination der Elemente von UML im selben Diagramm abbilden. Beispielsweise möchten Sie vielleicht sowohl Klassen als auch Objekte im gleichen Diagramm darstellen (dies ist gängige Praxis) oder Sie möchten vielleicht sogar Klassen und Komponenten im gleichen Diagramm darstellen (zulässig, aber nicht besonders gängig). Obwohl Sie nichts davon abhält, völlig unterschiedliche Modellierungsdinge zu einem Diagramm zusammenzufügen, ist es gängiger, in einem Diagramm ungefähr die gleiche Art von Dingen zusammenzufassen. Die definierten Diagramme von UML sind nach den Elementen benannt, die Sie am häufigsten in ihnen anordnen werden. Wenn Sie beispielsweise eine Menge von Klassen und ihre Beziehungen visualisieren möchten, werden Sie auf ein Klassendiagramm zugreifen. Wollen Sie allerdings eine Menge von Komponenten visualisieren, werden Sie auf ein Komponentendiagramm zugreifen.

### 7.1.2 Strukturdiagramme

Die Strukturdiagramme von UML dienen zur Visualisierung, Spezifizierung, Konstruktion und Dokumentation der statischen Aspekte eines Systems. Sie können sich die statischen Aspekte eines Systems als Abbildung des relativ stabilen Gerüsts bzw. Skeletts vorstellen. Ebenso wie die statischen Aspekte eines Hauses das Vorhandensein und die Anordnung solcher Dinge wie Wände, Türen, Fenster, Rohre, Leitungen und Belüftungen einschließen, umfassen die statischen Aspekte eines Softwaresystems das Vorhandensein und die Anordnung solcher Dinge wie Klassen, Interfaces, Kollaborationen, Komponenten und Knoten.

Die Strukturdiagramme von UML sind grob um die Hauptgruppen von Dingen organisiert, die Ihnen beim Modellieren eines Systems begegnen werden.

1. Klassendiagramm  
Klassen, Interfaces und Kollaborationen
2. Komponentendiagramm  
Komponenten
3. Montagediagramm  
Die interne Struktur
4. Objektdiagramm  
Objekte
5. Artefaktdiagramm  
Artefakte
6. Verteilungsdiagramm  
Knoten

*Klassendiagramme werden in Kapitel 8 behandelt.*

**Klassendiagramm** Ein Klassendiagramm stellt eine Menge von Klassen, Interfaces und Kollaborationen sowie ihre Beziehungen dar. Klassendiagramme sind die bei der Modellierung softwareintensiver Systeme am häufigsten eingesetzten Diagramme. Mithilfe von Klassendiagrammen können Sie die statische Entwurfs-sicht eines Systems darstellen. Klassendiagramme mit aktiven Klassen verwendet man zur Darstellung der statischen Prozess-sicht eines Systems.

*Montagediagramme und Komponentendiagramme werden in Kapitel 15 behandelt.*

**Komponentendiagramm** Ein *Komponentendiagramm* zeigt die internen Teile, Verbinder (Connectoren) und Ports, die eine Komponente implementieren. Bei Komponenteninstanzen sind die Kopien der internen Teile ebenfalls Instanzen.

**Montagediagramm** Ein *Montagediagramm* zeigt die interne Struktur einer Klasse oder einer Kollaboration. Der Unterschied zwischen Komponenten- und Montagestruktur ist gering, daher werden in diesem Buch beide als Komponentendiagramme behandelt.

*Objektdiagramme werden in Kapitel 14 behandelt.*

**Objektdiagramm** Ein *Objektdiagramm* zeigt eine Reihe von Objekten und deren Beziehungen. Sie setzen Objektdiagramme ein, um Datenstrukturen darzustellen, die statischen Schnappschüsse der Instanzen der in Klassendiagrammen zu findenden Dinge. Objektdiagramme bieten die statische Entwurfs-sicht oder die statische Prozess-sicht eines Systems ebenso wie es Klassendiagramme tun, allerdings aus einer Perspektive von realen Klassen oder Klassenprototypen.

**Artefaktdiagramm** Ein *Artefaktdiagramm* zeigt eine Menge von Artefakten und ihre Beziehungen zu anderen Artefakten sowie zu anderen Klassen, die sie implementieren. Sie verwenden Artefaktdiagramme, um die physischen Implementierungen eines Systems darzustellen. (Artefakte werden in UML zu den Verteilungsdiagrammen gezählt. Um deren Vorstellung zu vereinfachen, trennen wir diese allerdings.)

*Artefaktdiagramme werden in Kapitel 30 behandelt.*

**Verteilungsdiagramm** Ein *Verteilungsdiagramm* zeigt eine Menge von Knoten und deren Beziehungen. Mit Verteilungsdiagrammen stellen Sie die statische Verteilungssicht einer Architektur dar. Verteilungsdiagramme sind dadurch mit Komponentendiagrammen verwandt, dass ein Knoten üblicherweise eine oder mehrere Komponenten beinhaltet.

*Verteilungsdiagramme werden in Kapitel 31 behandelt.*

**Hinweis:** Es gibt einige geläufige Varianten dieser vier Diagramme, die nach ihrem Hauptzweck benannt sind. Beispielsweise können Sie ein Subsystemdiagramm erstellen, das die strukturelle Unterteilung eines Systems in Subsysteme darstellt. Ein Subsystemdiagramm ist nichts weiter als ein Klassendiagramm, das hauptsächlich Subsysteme darstellt.

## 7.2 Verhaltensdiagramme

Die Verhaltensdiagramme von UML dienen zur Visualisierung, Spezifizierung, Konstruktion und Dokumentation der dynamischen Aspekte eines Systems. Stellen Sie sich die dynamischen Aspekte eines Systems als seine veränderlichen Teile vor. Ebenso wie die dynamischen Aspekte eines Hauses die Luftzirkulation und das Verlassen und Betreten von Räumen beinhaltet, beinhalten die dynamischen Aspekte eines Softwaresystems solche Dinge wie den zeitlichen Verlauf von Nachrichten und den physischen Verlauf von Komponenten innerhalb eines Netzwerks. Die Verhaltensdinge von UML sind grob um die wichtigsten Methoden zur Modellierung der Dynamik eines Systems angeordnet.

### 1. Anwendungsfalldiagramm

Zur Organisation des Verhaltens eines Systems.

### 2. Sequenzdiagramm

Auf die zeitliche Abfolge von Nachrichten fokussiert.

### 3. Kommunikationsdiagramm

Auf den strukturellen Aufbau von Objekten fokussiert, die Nachrichten senden und empfangen.



#### 4. Zustandsdiagramm

Fokussiert auf den veränderlichen Zustand eines Systems, der durch Ereignisse hervorgerufen wird.

#### 5. Aktivitätsdiagramm

Auf den Steuerungsfluss von Aktivität zu Aktivität fokussiert.

*Anwendungsfall-  
diagramme wer-  
den in Kapitel 18  
behandelt.*

**Anwendungsfalldiagramm** Ein *Anwendungsfalldiagramm* zeigt eine Menge von Anwendungsfällen und Akteuren (eine besonderen Form von Klassen) und deren Beziehungen. Sie setzen Anwendungsfalldiagramme ein, um die statische Anwendungsfallsicht eines Systems darzustellen. Anwendungsfalldiagramme sind besonders für das Modellieren und Organisieren der Verhaltensweisen eines Systems von Bedeutung.

Sequenzdiagramme und Kommunikationsdiagramme werden unter dem Oberbegriff *Interaktionsdiagramme* zusammengefasst. Alle Sequenzdiagramme und Kommunikationsdiagramme sind Interaktionsdiagramme und ein Interaktionsdiagramm ist entweder ein Sequenzdiagramm oder ein Kommunikationsdiagramm. Diesen Diagrammen liegt dasselbe Modell zugrunde, obwohl beide in der Praxis unterschiedliche Dinge hervorheben. (Zeitdiagramme sind eine weitere Form der Interaktionsdiagramme, werden aber in diesem Buch nicht behandelt.)

*Sequenzdia-  
gramme werden  
in Kapitel 19  
behandelt.*

**Sequenzdiagramm** Ein *Sequenzdiagramm* ist ein Interaktionsdiagramm, das die zeitliche Abfolge von Nachrichten hervorhebt. Ein Sequenzdiagramm zeigt eine Reihe von Rollen sowie die Nachrichten, die von den die Rollen spielenden Instanzen gesendet und empfangen werden. Sequenzdiagramme setzt man ein, um die dynamische Sicht eines Systems darzustellen.

*Kommunikations-  
diagramme wer-  
den in Kapitel 19  
behandelt.*

**Kommunikationsdiagramm** Ein *Kommunikationsdiagramm* ist ein Interaktionsdiagramm, das die strukturelle Organisation der Dinge hervorhebt, die Nachrichten senden und empfangen. In einem Kommunikationsdiagramm sind Rollen, Verbinder (Connectoren) zwischen den Rollen und die Nachrichten dargestellt, die zwischen den die Rollen spielenden Instanzen ausgetauscht werden.

*Zustandsdia-  
gramme werden  
in Kapitel 25  
behandelt.*

**Zustandsdiagramm** Ein *Zustandsdiagramm* stellt einen Zustandsautomaten dar, der aus Zuständen, Übergängen, Ereignissen und Aktivitäten besteht. Zustandsdiagramme werden eingesetzt, um die dynamische Sicht eines Systems zu modellieren. Sie sind vor allem für das Modellieren eines Interface, einer Klasse oder einer Kollaboration von Bedeutung. Bei Zustandsdiagrammen liegt der Schwerpunkt auf dem Verhalten eines Objekts nach einer Reihenfolge von Ereignissen. Dies ist für das Modellieren reaktiver Systeme von besonderer Bedeutung.

**Aktivitätsdiagramme** Ein *Aktivitätsdiagramm* zeigt den schrittweisen Verlauf einer Berechnung.

Eine Aktivität stellt eine Reihe von Aktionen, den verzweigenden oder linearen Verlauf von Aktion zu Aktion und die Werte, die von den Aktionen rückgemeldet oder benötigt werden. Mit Aktivitätsdiagrammen stellen Sie die dynamische Sicht eines Systems dar. Aktivitätsdiagramme sind besonders für die Modellierung der Funktion eines Systems von Bedeutung. Aktivitätsdiagramme unterstreichen den Steuerungsfluss innerhalb der Ausführung eines Verhaltens.

*Aktivitätsdiagramme, eine besondere Form der Zustandsdiagramme, werden in Kapitel 20 behandelt.*

**Hinweis:** Die Darstellung aller dynamischen Sachverhalte (beispielsweise des Verhaltens eines Systems) in Form von Diagrammen (an sich statische Artefakte, vor allem, wenn sie auf Papier, einer Tafel oder der Rückseite eines Umschlags gezeichnet sind) unterliegt offensichtlichen praktischen Einschränkungen. Werden sie auf einem Computerbildschirm dargestellt, gibt es die Möglichkeit, Verhaltensdiagramme zu animieren, so dass sie entweder ein ausführbares System simulieren oder ein System widerspiegeln, das sich gerade in der Ausführung befindet. UML erlaubt es, dynamische Diagramme zu erzeugen und Farbe und weitere visuelle Hilfsmittel einzusetzen, um das Diagramm »laufen zu lassen«. Einige Tools bieten bereits diese fortgeschrittene Anwendung von UML.

## 7.3 Modellierungstechniken

### 7.3.1 Verschiedene Sichten eines Systems modellieren

Wenn Sie ein System aus verschiedenen Sichten modellieren, bauen Sie Ihr System in Wirklichkeit von unterschiedlichen Dimensionen ausgehend neu auf. Indem Sie eine geeignete Menge von Sichten auswählen, setzen Sie einen Prozess in Gang, der Sie zwingt, sich geeignete Fragen über Ihr System zu stellen und Risiken offen zu legen, die angegangen werden müssen. Wenn Sie bei der Auswahl dieser Sichten falsche Entscheidungen treffen oder Sie sich auf Kosten aller anderen auf eine Sicht konzentrieren, laufen Sie Gefahr, Schwierigkeiten zu verbergen und Probleme zurückzustellen, was am Ende jede Erfolgsaussicht zunichte macht.

Um ein Modell aus verschiedenen Sichten zu modellieren,

- ▶ entscheiden Sie, welche Sichten Sie benötigen, um die Architektur Ihres Systems bestmöglich darzustellen und die technischen Risiken, die sich für Ihr Projekt stellen, hervorzuheben. Die bereits beschriebenen fünf Sichten einer Architektur sind ein hervorragender Ausgangspunkt.
- ▶ entscheiden Sie, welche Artefakte Sie für jede dieser fünf Sichten erzeugen müssen, um die wichtigen Details dieser Sicht

ausdrücken zu können. Diese Artefakte bestehen größtenteils aus verschiedenen UML-Diagrammen.

- ▶ nehmen Sie die Entscheidung, welche dieser Diagramme Sie auf eine formale oder teilweise formale Weise kontrollieren wollen, in Ihre Prozessplanung auf. Dies sind die Diagramme, für die Sie Überarbeitungen einplanen müssen und die Sie als Dokumentation des Projekts erhalten müssen.
- ▶ planen Sie Diagramme ein, die am Ende verworfen werden. Solche Übergangsdigramme sind geeignet, um die Auswirkungen von Entscheidungen zu erforschen und um mit Veränderungen zu experimentieren.

Wenn Sie beispielsweise eine monolithische Anwendung (die nur aus einem großen Block besteht) modellieren, die auf einem Einzelplatzrechner läuft, benötigen Sie möglicherweise nur eine Hand voll Diagramme.

- ▶ Anwendungsfallsicht – Anwendungsfalldiagramme
- ▶ Entwurfssicht – Klassendiagramme (Strukturmodellierung)
- ▶ Interaktionssicht – Interaktionsdiagramme (Verhaltensmodellierung)
- ▶ Implementierungssicht Montagediagramme
- ▶ Verteilungssicht – nicht erforderlich

Besitzt das von Ihnen modellierte System reaktive Eigenschaften oder konzentriert es sich auf die Abläufe innerhalb eines Vorgangs, empfiehlt es sich, für eine bessere Modellierung Ihres Systems Zustandsdiagramme und Aktivitätsdiagramme hinzuzufügen.

Wenn es sich um ein Client-Server-System handelt, empfiehlt es sich, Komponentendiagramme und Verteilungsdiagramme einzusetzen, um die physikalischen Einzelheiten Ihres Systems zu modellieren.

Wenn Sie schließlich ein umfangreiches, verteiltes System modellieren, benötigen Sie, wie im Folgenden aufgeführt, alle Diagramme, die UML zu bieten hat, um die Architektur und die technischen Risiken des betroffenen Systems ausdrücken zu können.

- ▶ Anwendungsfallsicht – Anwendungsfalldiagramme  
Sequenzdiagramme
- ▶ Entwurfssicht – Klassendiagramme (Strukturmodellierung)  
Interaktionsdiagramme (Verhaltensmodellierung)  
Zustandsdiagramme (Verhaltensmodellierung)  
Aktivitätsdiagramme

- ▶ Interaktionssicht – Interaktionsdiagramme (Verhaltensmodellierung)
- ▶ Implementierungssicht – Klassendiagramme  
Montagediagramme
- ▶ Verteilungssicht – Verteilungsdiagramme

### 7.3.2 Verschiedene Abstraktionsebenen modellieren

Sie müssen nicht nur ein System aus verschiedenen Blickwinkeln betrachten, es wird ebenfalls vorkommen, dass an der Entwicklung beteiligte Personen dieselbe Ansicht des Systems benötigen, allerdings auf verschiedenen Abstraktionsebenen. Ein Programmierer beispielsweise könnte bei einer Menge von Klassen, die das Vokabular des Problemraums ausdrücken, eine detaillierte Ansicht auf einer Ebene benötigen, in der die Attribute, Operationen und Beziehungen jeder einzelnen Klasse dargestellt sind. Ein Analyst, der mit einem Anwender einige Anwendungsfallszenarien durcharbeitet, benötigt sicherlich keine derart detaillierte Ansicht dieser Klassen. Man spricht in diesem Zusammenhang davon, dass der Programmierer auf einer tieferen Abstraktionsebene arbeitet, während Analyst und Anwender auf einer höheren Abstraktionsebene arbeiten. Beide arbeiten jedoch am gleichen Modell. Da ein Diagramm in Wirklichkeit nichts weiter ist als eine Darstellung der Elemente, aus denen ein Modell aufgebaut ist, können Sie von einem Modell oder mehreren Modellen die verschiedensten Diagramme herleiten, in denen jeweils bestimmte Anordnungen der Elemente aus- oder eingeblendet werden, und die verschiedenen Detailstufen darstellen.

Es gibt zur Modellierung eines Systems auf verschiedenen Abstraktionsebenen zwei Vorgehensweisen: durch Erzeugen von Diagrammen mit verschiedenen Abstraktionsebenen anhand desselben Modells oder durch Erzeugen von Modellen mit verschiedenen Abstraktionsebenen mit Diagrammen, die von einem Modell zum nächsten verlaufen.

Um ein Modell auf verschiedenen Abstraktionsebenen durch die Darstellung von Diagrammen mit verschiedenen Detailstufen zu modellieren,

- ▶ beginnen Sie mit einem bestimmten Modell und achten Sie auf die Anforderungen des Lesers.
- ▶ Wenn Ihr Leser das Modell benutzt, um damit eine Implementierung zu erzeugen, muss es sich auf einer niedrigen Abstraktionsebene befinden, es müssen also viele Details dargestellt werden. Wird das Modell eingesetzt, um einem Kunden ein konzeptuelles Modell zu erklären, sind Diagramme erforder-

lich, die sich auf einer höheren Abstraktionsebene befinden, die also kaum Details darstellen.

- ▶ Erzeugen Sie in Abhängigkeit davon, wo Sie sich am Ende in diesem Spektrum von hohen bis niedrigen Abstraktionsebenen befinden, ein Diagramm mit einer geeigneten Abstraktionsebene, indem Sie in Ihrem Modell die folgenden vier Kategorien von Dingen aus- oder einblenden:

*Nachrichten werden in Kapitel 16 behandelt, Übergänge sind das Thema in Kapitel 22, Stereotypen werden in Kapitel 6 behandelt.*

1. *Bausteine und Beziehungen*: Blenden Sie diejenigen aus, die für den beabsichtigten Verwendungszweck Ihres Diagramms bzw. die Bedürfnisse der Person, die das Diagramm verwenden wird, nicht relevant sind.
2. *Zusätze*: Bilden Sie nur die Zusätze dieser Bausteine und Beziehungen ab, die für das Verständnis des beabsichtigten Inhalts notwendig sind.
3. *Fluss*: Bilden Sie in Zusammenhang von Verhaltensdiagrammen nur solche Meldungen oder Übergänge ab, die für das Verständnis des beabsichtigten Inhalts notwendig sind.
4. *Stereotype*: Bilden Sie in Zusammenhang von Stereotypen, die zur Klassifizierung einer Liste von Dingen, beispielsweise Attributen und Operationen, eingesetzt werden, nur solche Stereotypen ab, die für das Verständnis des beabsichtigten Inhalts notwendig sind.

Der Hauptvorteil dieser Herangehensweise liegt darin, dass Sie stets auf einen allgemeinen semantischen Speicher (Repository) zugreifen. Der Hauptnachteil dieser Herangehensweise hingegen ist, dass Änderungen an einem Diagramm auf einer bestimmten Abstraktionsebene dazu führen, dass die Diagramme auf einer anderen Abstraktionsebene nicht mehr aktuell sind.

Um ein Modell auf verschiedenen Abstraktionsebenen durch die Erstellung von Modellen auf verschiedenen Detailstufen zu modellieren,

- ▶ untersuchen Sie die Anforderungen der Leser und entscheiden Sie sich für die Abstraktionsebene, die jeder Leser benötigt. Erstellen Sie dann für jede Ebene ein Modell.
- ▶ füllen Sie Ihre Modelle, die sich auf einer hohen Abstraktionsebene befinden, mit einfachen Abstraktionen und diejenigen Modelle, die sich auf einer niedrigen Abstraktionsebene befinden, mit detaillierteren Abstraktionen. Erstellen Sie Ablaufabhängigkeiten zwischen den verwandten Elementen verschiedener Modelle.
- ▶ Wenn Sie die fünf Ansichten einer Architektur befolgen, werden Ihnen in der Praxis des Modellierens eines Systems auf verschiedenen Abstraktionsebenen die folgenden vier Situationen häufig begegnen:

*Ablaufabhängigkeiten werden in Kapitel 32 behandelt.*

1. *Anwendungsfälle und ihre Umsetzung*: Anwendungsfälle in einem Anwendungsfalldiagramm lassen sich auf Kollaborationen in einem Entwurfsmodell zurückverfolgen.
2. *Kollaborationen und ihre Umsetzung*: Kollaborationen lassen sich auf eine Reihe von Klassen zurückverfolgen, die zur Durchführung der Kollaboration zusammenwirken.
3. *Komponenten und ihre Entwürfe*: Komponenten in einem Implementierungsmodell lassen sich auf die Elemente in einem Entwurfsmodell zurückverfolgen.
4. *Knoten und ihre Komponenten*: Knoten in einem Verteilungsmodell lassen sich auf die Komponenten in einem Implementierungsmodell zurückverfolgen.

Der Hauptvorteil dieser Herangehensweise liegt darin, dass die Diagramme auf unterschiedlichen Abstraktionsebenen weniger eng verknüpft sind, Änderungen an einem Modell sich also weniger direkt auf andere Modelle auswirken. Der Hauptnachteil dieser Herangehensweise wiederum liegt darin, dass für das Erhalten dieser Modelle auf gleichem Stand Ressourcen erforderlich sind. Dies gilt besonders, wenn Ihre Modelle parallel zu unterschiedlichen Phasen des Lebenszyklus der Softwareentwicklung verlaufen, wenn Sie sich beispielsweise dazu entschließen, ein Analysemodell getrennt von einem Entwurfsmodell zu erhalten.

*Interaktionsdiagramme werden in Kapitel 19 behandelt.*

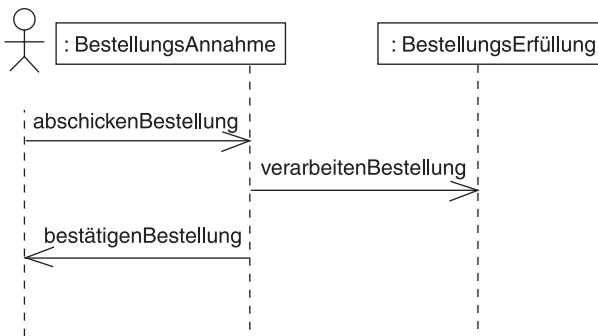


Abbildung 7.1: Interaktionsdiagramm auf einer hohen Abstraktionsebene

Stellen Sie sich beispielsweise einmal vor, dass Sie ein System eines Internetversands modellieren. Einer der wichtigsten Nutzfälle eines solchen Systems wäre das Bestellen. Wenn Sie ein Analyst oder Anwender sind, erzeugen Sie wahrscheinlich, wie in Abbildung 7.1 gezeigt, einige Interaktionsdiagramme auf einer hohen Abstraktionsebene, die den Vorgang des Bestellens abbilden.

Ein Programmierer hingegen, der für die Implementierung dieses Szenarios verantwortlich ist, würde dieses Diagramm um einige

Nachrichten und weitere Teilnehmer erweitern, wie in Abbildung 7.2 gezeigt.

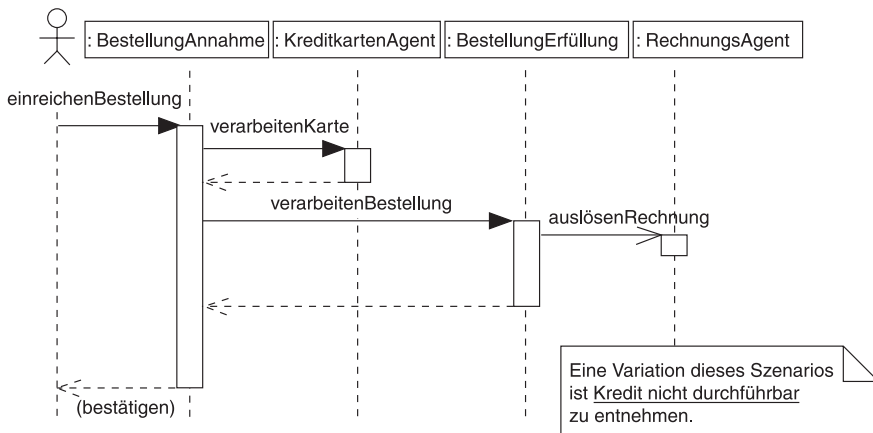


Abbildung 7.2: Interaktion auf einer niedrigen Abstraktionsebene

Beide Diagramme beziehen sich auf dasselbe Modell, sie befinden sich allerdings auf verschiedenen Abstraktionsebenen. Das zweite Diagramm verfügt über zusätzliche Nachrichten und Rollen. Die Verwendung vieler derartiger Diagramme empfiehlt sich aber besonders dann, wenn Ihre Tools eine einfache Navigation von einem Diagramm zu einem anderen erlauben.

### 7.3.3 Komplexe Sichten modellieren

Auch wenn Sie Ihre Modelle in kleine Einheiten unterteilen, kommt es vor, dass Sie große und umfangreiche Diagramme erstellen müssen. Wenn Sie beispielsweise das gesamte Schema einer Datenbank analysieren möchten, das 100 oder mehr Abstraktionen umfasst, lohnt es sich wirklich, ein Diagramm zu untersuchen, das all diese Klassen und ihre Kollaborationen enthält. Wenn Sie dies tun, werden Sie allgemeine Kollaborationsmuster erkennen. Würden Sie dieses Modell durch Ausblenden bestimmter Details auf einer höheren Abstraktionsebene darstellen, wären die für diese Einsichten erforderlichen Informationen nicht mehr enthalten.

Um komplexe Sichten zu modellieren,

*Pakete werden  
in Kapitel 12  
behandelt, Kolla-  
borationen in  
Kapitel 28.*

- ▶ stellen Sie zuerst sicher, dass sich diese Informationen nicht sinnvoll auf einer höheren Abstraktionsebene darstellen lassen, beispielsweise durch Ausblenden einiger Teile des Diagramms und Erhalten der Details in anderen Teilen.
- ▶ Wenn Sie alle möglichen Details ausgeblendet haben und das Diagramm weiterhin zu umfangreich ist, erwägen Sie, einige

Elemente in Paketen oder in Kollaborationen auf einer höheren Ebene zusammenzufassen. Stellen Sie in Ihrem Diagramm dann nur diese Pakete oder Kollaborationen dar.

- ▶ Ist Ihr Diagramm weiterhin zu umfangreich, setzen Sie Notizen und Farben als visuelle Hilfsmittel ein, um die Aufmerksamkeit des Lesers auf die beabsichtigten Aussagen zu lenken.
- ▶ Ist Ihr Diagramm weiterhin zu umfangreich, drucken Sie das gesamte Dokument aus und hängen Sie es an eine entsprechend große Wand. Dabei geht die rechnergestützte Interaktivität des Diagramms zwar verloren, Sie können das gesamte Diagramm nun jedoch aus einer etwas größeren Entfernung nach allgemeinen Mustern durchsuchen.

## 7.4 Hinweise und Tipps

Beim Erstellen eines Diagramms,

- ▶ denken Sie daran, dass die Erstellung eines Diagramms in UML nicht den Zweck hat, schöne Bilder zu malen, sondern zum Visualisieren, Spezifizieren, Konstruieren und Dokumentieren dient. Diagramme sind ein Mittel zum Zweck und der Zweck ist die Bereitstellung eines ausführbaren Systems.
- ▶ Nicht alle Diagramme müssen aufgehoben werden. Erwägen Sie, Modelle nebenbei zu erzeugen, indem Sie die Elemente in Ihren Modellen hinterfragen. Setzen Sie diese Modelle dann ein, um über das System nachzudenken, das sich im Bau befindet. Viele Diagramme dieses Typs können verworfen werden, nachdem sie ihren Zweck erfüllt haben. (Die Semantik, auf der sie basieren, besteht allerdings weiterhin als Teil des Modells.)
- ▶ Vermeiden Sie irrelevante Diagramme oder redundante Darstellungen. Diese überfüllen nur Ihre Modelle.
- ▶ Stellen Sie in jedem Diagramm nur die für den Einsatzzweck relevanten Details dar. Zusätzliche irrelevante Informationen können den Leser von der beabsichtigten Kernaussage ablenken.
- ▶ Stellen Sie andererseits keine allzu minimalistischen Diagramme her, wenn Sie nicht einen Sachverhalt auf einer äußerst hohen Abstraktionsebene darstellen wollen.
- ▶ Schaffen Sie ein Gleichgewicht zwischen den Strukturdiagrammen und den Verhaltensdiagrammen in Ihrem System. Nur wenige Systeme sind rein statisch oder rein dynamisch.
- ▶ Erstellen Sie Diagramme, die weder zu groß (sind sie größer als eine ausgedruckte Seite, sind sie schwer zu lesen) noch zu



klein sind. (Erwägen Sie, mehrere triviale Diagramme zu einem zusammenzufügen.)

- ▶ Geben Sie jedem Diagramm einen aussagekräftigen Namen, der die dahinter stehende Absicht deutlich ausdrückt.
- ▶ Organisieren Sie Ihre Diagramme. Gruppieren Sie sie entsprechend der Ansicht zu Paketen.
- ▶ Befassen Sie sich nicht unnötig lange mit dem Format eines Diagramms. Lassen Sie sich von Tools helfen.

Ein gut strukturiertes Diagramm

- ▶ konzentriert sich darauf, einen Aspekt der Ansicht eines Systems zu übermitteln.
- ▶ enthält nur solche Elemente, die für das Verständnis dieses Aspekts unerlässlich sind.
- ▶ bietet eine der Abstraktionsebene entsprechende Detailstufe (stellen Sie nur solche Zusätze dar, die für das Verständnis unbedingt erforderlich sind).
- ▶ ist nicht derart minimalistisch, dass es Leser über wichtige Aspekte der Semantik fehlinformiert.

Beim Erstellen eines Diagramms

- ▶ geben Sie diesem einen Namen, an dem der Einsatzzweck erkennbar ist.
- ▶ ordnen Sie die Elemente so an, dass überkreuzende Linien weitestgehend vermieden werden.
- ▶ ordnen Sie die Elemente räumlich so an, dass semantisch verwandte Dinge nahe beieinander liegen.
- ▶ setzen Sie Farben und Notizen als visuelle Hinweise ein, um die Aufmerksamkeit auf wichtige Eigenschaften Ihres Diagramms zu lenken. Gehen Sie beim Einsatz von Farben jedoch vorsichtig vor, da es viele farbenblinde Menschen gibt. Farbe sollte nur zur Hervorhebung eingesetzt werden, nicht zum Übermitteln wichtiger Informationen.