

## 24 Transformation der Anforderungsspezifikation

### 24.1 Einleitung

Bei der Softwarespezifizierung wird die Anforderungsspezifikation überarbeitet, weiter strukturiert und präzisiert, um eine Basis für die Realisierung zu erhalten. In einem ersten Schritt werden die Anwendungsfälle zusammen mit dem Domänenklassenmodell der Anforderungsspezifikation in das initiale Klassenmodell der Softwarespezifikation transformiert. Nicht zuletzt aus Gründen der Verfolgbarkeit sollten sich das Domänenklassenmodell und die Anwendungsfälle dabei im resultierenden Klassenmodell so weit wie möglich strukturell wiederfinden.

Trotz ihrer gewissen »Realisierungsähnlichkeit«, die der Zielgruppe Entwickler entgegenkommt, stellen die Klassen der Softwarespezifikation hinsichtlich folgender Aspekte immer noch absolut plattformunabhängige Abstraktionen von Teilsystemen, Komponenten oder Klassen des Entwurfs oder der Implementierung dar [JBR99]:

- Die Klassen der Softwarespezifikation fokussieren auf funktionale Anforderungen; bei der Realisierung müssen zusätzlich auch die nichtfunktionalen Anforderungen berücksichtigt werden.
- Attribute dieser Klassen werden möglichst problemnah und noch relativ abstrakt beschrieben. Nur in Ausnahmefällen werden Typen auf programmiersprachlicher Ebene angegeben.
- Beziehungen im Klassenmodell der Softwarespezifikation abstrahieren von der Realisierung, indem z.B. im Falle von Assoziationen keine Navigierbarkeit angegeben wird. Generalisierung wird nur zur konzeptionellen Strukturierung und nicht zur reinen Redundanzvermeidung eingesetzt.
- Das Verhalten (von Instanzen) der Klassen wird in der Softwarespezifikation soweit möglich in Form von Verantwortlichkeiten (responsibilities, vgl. Abschnitt 11.3) und Schnittstellen (Abschnitt 10.1) skizziert. Jede Verantwortlichkeit umreißt einen inhaltlich zusammenhängenden Teil der Schnittstelle einer Klasse.
- Bei der Angabe von (Nicht-Standard-)Operationen besteht die Gefahr, zu stark in Realisierungskategorien zu denken. Oft reichen zur Beschreibung einer Klasse in der Softwarespezifikation bestimmte Standardoperationen wie z.B. Zugriff auf Attribute oder Einfügen, Suchen und Entfernen von Verbindungen aus, die ohne explizite Definition zur Verfügung stehen. Darüber hinausgehende Operationen

werden nur aufgenommen, wenn sie zum Verständnis der Verantwortlichkeit der zugehörigen Klasse notwendig sind.

- Präzise Schnittstellenbeschreibungen mit Signaturen werden nur für komplexe (Nicht-Standard-)Operationen angegeben. Klasseninvarianten sowie Vor- und Nachbedingungen komplexer Operationen präzisieren hierbei ausschließlich die (umgangssprachlich formulierten) Vor- und Nachbedingungen der Anwendungsfälle.

Zur besseren Unterscheidung hinsichtlich ihrer grundsätzlichen Verantwortlichkeit werden die Klassen der Softwarespezifikation einer der drei Kategorien *Entitätsklasse*, *Anwendungsfall-Funktionsklasse* (AF-Klasse) und *Geschäftsprozess-Funktionsklasse* (GF-Klasse) zugeordnet:

- *Entitätsklassen* repräsentieren Anwendungsfall- und Geschäftsprozessfunktionsübergreifende, langlebige Informationen des Anwendungssystems,
- *Anwendungsfall-Funktionsklassen* sind verantwortlich für die anwendungsfallbezogene fachliche Logik sowie die dafür notwendigen Interaktionen des Anwendungssystems mit seinen Akteuren, und
- *Geschäftsprozess-Funktionsklassen* sind verantwortlich für die auf Funktionen der Geschäftsprozesse bezogene fachliche Logik sowie die entsprechenden Interaktionen des Anwendungssystems mit seinen Akteuren. Sie werden nur benötigt, wenn das Anwendungssystem nicht in ein übergeordnetes WFMS integriert wird.

Um diese drei Kategorien von Klassen in der Softwarespezifikation augenfällig ausdrücken zu können, werden drei Stereotype «Entitätsklasse», «AF-Klasse» und «GF-Klasse» definiert, deren zugehörige Symbole Abbildung 24–1 zeigt. Aufgrund ihrer ähnlichen Verantwortlichkeiten werden AF- und GF-Klassen mit ähnlichen Symbolen dargestellt.



**Abb. 24–1** Symbole der drei Klassen-Stereotype der Softwarespezifikation

In Klassendiagrammen der Softwarespezifikation werden oft lediglich diese Symbole benutzt. Will man jedoch die Eigenschaften einer solchen Klasse genauer darstellen, kann man je nach Geschmack und Bedarf eine der anderen in Abbildung 24–2 am Beispiel einer Entitätsklasse gezeigten Darstellungsvarianten auswählen.

In den nächsten Abschnitten wird die Transformation der Elemente der Anforderungsspezifikation in die drei Kategorien von Klassen der Softwarespezifikation im Einzelnen besprochen. Normalerweise transformiert man zunächst die Domänenklassen, dann die Anwendungsfälle und danach ggf. die übergeordneten GF-Anwendungsfälle.

## 24.2 Domänenklassen → Entitätsklassen

Die Übertragung von Domänenklassen in das Klassenmodell der Softwarespezifikation kann vergleichsweise schematisch erfolgen, denn Entitätsklassen repräsentieren ja langlebige Information im Anwendungssystem und müssen damit auf jeden Fall alle Domänenklassen der Anforderungsspezifikation widerspiegeln.

Domänenklassen werden also mit dem Stereotyp «Entitätsklasse» oder dem entsprechenden Symbol gekennzeichnet (vgl. Abb. 24–2) und zusammen mit ihren Beziehungen als Klassen in die Softwarespezifikation übernommen. Im weiteren Verlauf der Softwarespezifizierung werden Attribute und Assoziationen der Entitätsklassen noch verfeinert sowie erstmalig auch Nicht-Standardoperationen aufgenommen. Grundsätzlich findet sich das Domänenklassenmodell also weitgehend strukturgleich im Klassenmodell der Softwarespezifikation wieder.

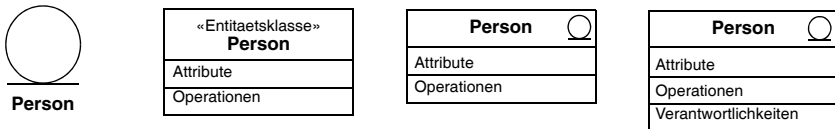


Abb. 24–2 Darstellungsvarianten einer Entitätsklasse

**Beispiel 24–1** Abbildung 24–3 zeigt die Transformation der Domänenklassen *Bestellung* und *Kunde* in das initiale Klassenmodell der Softwarespezifikation des Bestellwesen-Beispiels. Diese Klassen inkl. ihrer Attribute und der Assoziationen wurden aus dem Domänen-Klassendiagramm übernommen und mit dem Stereotyp «Entitätsklasse»

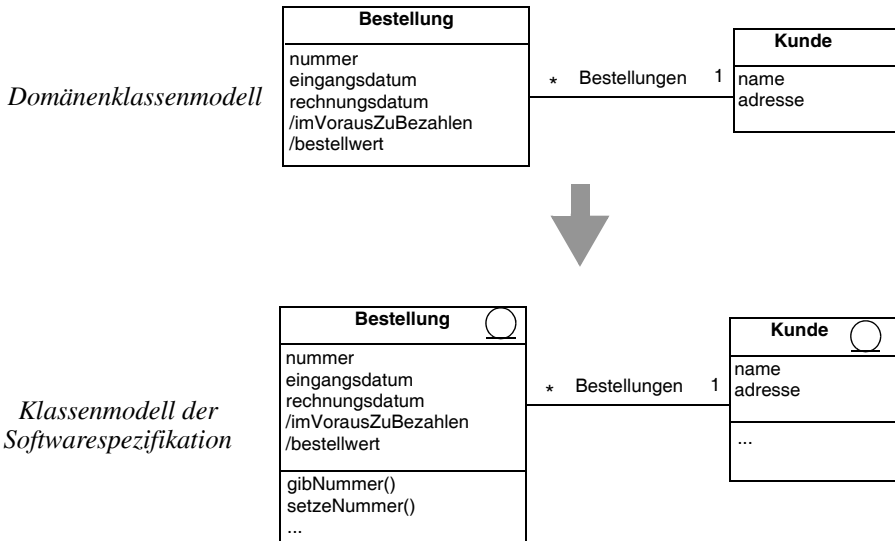
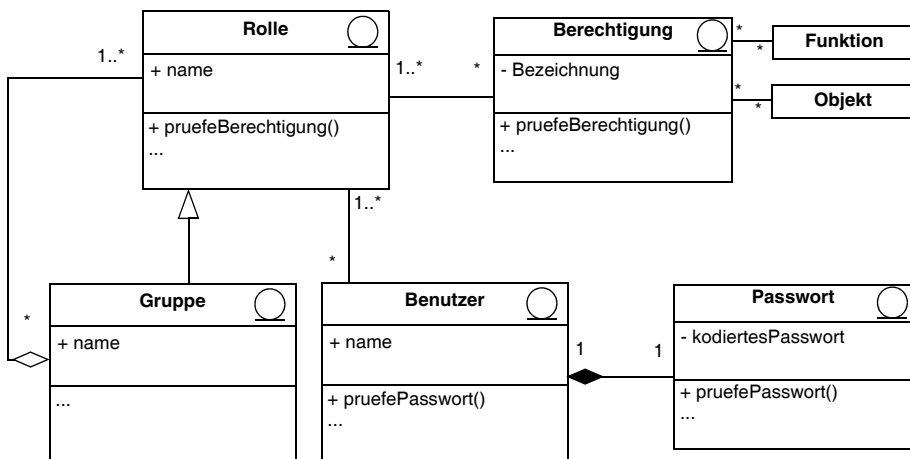


Abb. 24–3 Transformation zweier Domänenklassen in Entitätsklassen im Bestellwesen

gekennzeichnet. Im weiteren Verlauf werden diesen Klassen dann Operationen und ggf. weitere Attribute hinzugefügt.

Zusätzlich werden oft noch weitere, eher technisch motivierte, systembezogene Klassen als Entitätsklassen aufgenommen. Diese ergeben sich aus solchen Anforderungen, die nicht unmittelbar der Domäne entstammen, sondern erst im Rahmen des Systemeinsatzes erforderlich werden, wie z.B. Identifizierung und Authentifizierung der Benutzer im Rahmen von Zugangsbeschränkungen oder Funktionen zur Systemadministration.

**Beispiel 24–2** Die in Abbildung 24–4 dargestellten Entitätsklassen Rolle, Gruppe, Berechtigung, Benutzer und Passwort dienen als Beispiel für zusätzlich benötigte, eher systembezogene Klassen. Sie sind dem Anwendungsfall »Systemanmeldung« aus dem Anwendungsfalldiagramm der Anforderungsspezifikation (vgl. Abb. 12–13) zugeordnet. Jeder Benutzer hat ein Passwort und gehört zu einer oder mehreren (Gruppen von) Rollen, denen jeweils bestimmte Berechtigungen zum Zugriff auf Funktionen und Objekte des Anwendungssystems zugewiesen sind.



**Abb. 24–4** »Systembezogene« Entitätsklassen zur rollenbasierten Berechtigung

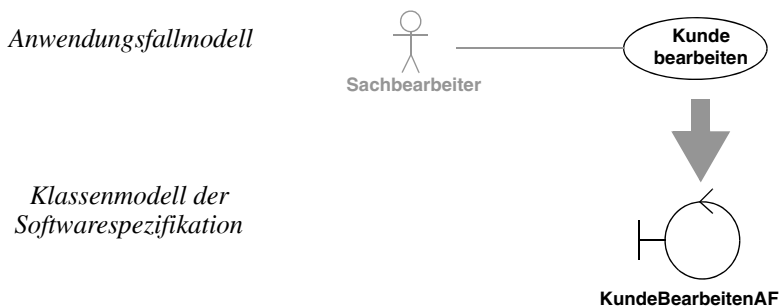
## 24.3 Anwendungsfälle → AF-Klassen

Nach der relativ einfachen, da weitgehend strukturerhaltenden Überführung von Domänenklassen in Entitätsklassen folgt jetzt die Transformation des Anwendungsfalldiagramms. Der weitere Verlauf der Softwarespezifizierung konzentriert sich ausschließlich auf die Belange der menschlichen Akteure, deren Handlungen durch Anwendungsfälle des Anwendungssystems zu unterstützen sind. Für technische bzw. maschinelle Akteure sind in der Softwarespezifikation die Protokolle bzw. Schnittstellen präzise zu

beschreiben. Dies kann entweder textuell erfolgen oder aber durch entsprechende Interaktions- und/oder Zustandsdiagramme. Weiter gehende Betrachtungen der technischen Akteure finden erst während der Realisierung statt (vgl. Abschnitt 35.2).

In der Softwarespezifikation geht man davon aus, dass (insbesondere menschliche) Akteure die ihnen zur Verfügung stehenden Anwendungsfälle »irgendwie« aktivieren können. Ob dies z.B. über ein Auswahlmenü in einem Fenster oder über entsprechende Tastaturbefehle erfolgt, wird hier nicht weiter spezifiziert. Erste derartige Angaben wurden ja bereits in der Anforderungsermittlung in Form von Bildschirmsskizzen angefertigt (vgl. Kapitel 19). Hat der Akteur den von ihm gewünschten Anwendungsfall ausgewählt und aktiviert, ist der weitere Ablauf abhängig vom ausgewählten Anwendungsfall. Da sich die Abläufe unterschiedlicher Anwendungsfälle normalerweise stark unterscheiden, wird jeder Anwendungsfall auf eine *Anwendungsfall-Funktionsklasse* (AF-Klasse) abgebildet, deren Attribute und Operationen die anwendungsfallbezogene Interaktion des Systems mit dem Akteur bzw. den Akteuren spezifizieren. Eine AF-Klasse wird durch den Namen des Anwendungsfalls mit angehängtem AF bezeichnet. Ist der resultierende Name der AF-Klasse zu lang, können unter Beibehaltung des Suffixes AF geeignete Kürzel gewählt werden.

**Beispiel 24–3** Der Anwendungsfall »Kunde bearbeiten« des Bestellwesens wird vom Akteur Sachbearbeiter aktiviert. Abbildung 24–5 zeigt die resultierende Anwendungsfall-Funktionsklasse KundeBearbeitenAF.



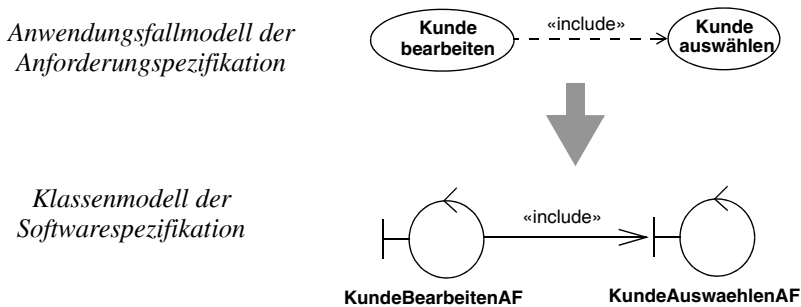
**Abb. 24–5** Anwendungsfall-Funktionsklasse (AF-Klasse) im Bestellwesen

Neben der Interaktion mit den Akteuren wird auch die den Anwendungsfällen zugrunde liegende fachliche Logik in der AF-Klasse gekapselt. Zu den diesbezüglichen Verantwortlichkeiten der Instanzen einer AF-Klasse gehört hauptsächlich die Kontrolle der an dem Anwendungsfall beteiligten (Instanzen von) Entitätsklassen gemäß der Anforderungsspezifikation. AF-Klassen spielen damit eine Art Vermittlerrolle zwischen den Akteuren und den Instanzen von Entitätsklassen. Später werden ihnen noch Details bezüglich der Benutzungsschnittstelle hinzugefügt (siehe Kapitel 26).

Die Transformation der include- und extend-Beziehungen des Anwendungsfallmodells ergibt einseitig navigierbare, stereotypisierte Assoziationen zwischen den aus den Anwendungsfällen resultierenden AF-Klassen. Die Navigierbarkeit dieser Assozi-

ationen ist bei der Transformation einer include-Beziehung von der AF-Klasse des Basisanwendungsfalls hin zu der des benutzten Anwendungsfalls gerichtet, bei der Transformation einer extend-Beziehung verläuft sie von der AF-Klasse des erweiternden Anwendungsfalls zu der des Basisanwendungsfalls.

**Beispiel 24–4** Im Anwendungsfallmodell der Anforderungsspezifikation unterhält der Anwendungsfall »Kunde bearbeiten« eine include-Beziehung zum Anwendungsfall »Kunde auswählen«, damit der Akteur Sachbearbeiter eine Instanz der Klasse Privatkunde auswählen kann, um sie ggf. zu modifizieren. Abbildung 24–6 zeigt, wie diese include-Beziehung in eine einseitig navigierbare Assoziation im Klassenmodell der Softwarespezifikation transformiert wird.

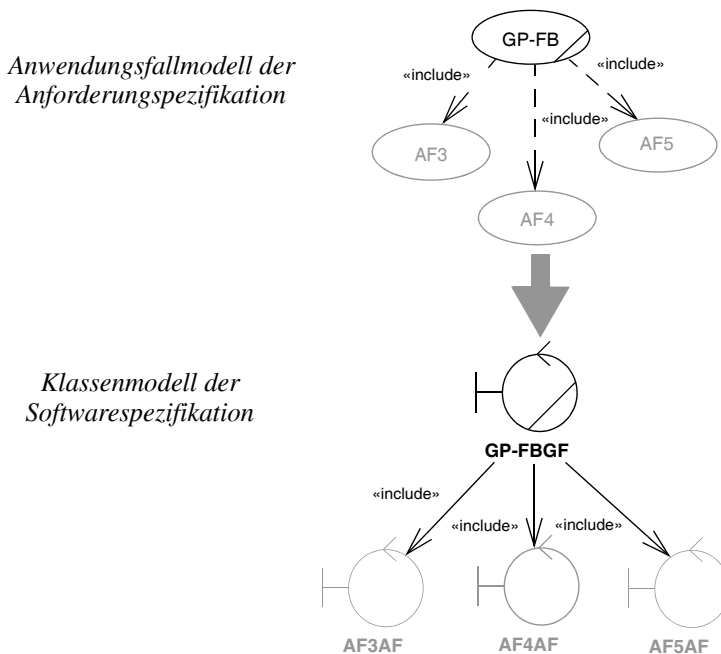


**Abb. 24–6** Übertragung einer include-Beziehung vom Anwendungsfallmodell in das Klassenmodell der Softwarespezifikation

## 24.4 GF-Anwendungsfälle → GF-Klassen

Ist das Anwendungssystem nicht in ein übergeordnetes WFMS integriert, wurden die komplexen Funktionen der Geschäftsprozesse mit GF-Anwendungsfällen in der Anforderungsspezifikation berücksichtigt (Abschnitt 17.4). Diese werden nun in der Softwarespezifikation mit jeweils einer *Geschäftsprozess-Funktionsklasse* (GF-Klasse) ausgedrückt. Eine GF-Klasse wird durch den Namen des GF-Anwendungsfalls mit angehängtem GF bezeichnet. Von der GF-Klasse werden einseitig navigierbare stereotypisierte Assoziationen zu den AF-Klassen der in der Geschäftsprozessfunktion benötigten Anwendungsfälle eingefügt.

**Beispiel 24–5** Dem GF-Anwendungsfall »GP-FB« sind die Anwendungsfälle »AF3« bis »AF5« zugeordnet, so dass er mit diesen im Anwendungsfallmodell über include-Beziehungen verbunden ist (vgl. Beispiel 17–4). Abbildung 24–7 zeigt, wie er in die Klasse GP-FBGF der Softwarespezifikation transformiert und aufgrund der include-Beziehungen über entsprechend stereotypisierte Assoziationen mit den AF-Klassen verknüpft wird.



**Abb. 24–7** Übertragung des GF-Anwendungsfalls »GP-FB« in das Klassenmodell der Softwarespezifikation

## 24.5 Abläufe und Interaktionsdiagramme → Operationen

Um die drei Blickwinkel Funktion, Struktur und Verhalten der Anforderungsspezifikation nicht zu verwischen, wurde bei der Anforderungsermittlung bewusst auf die Modellierung von Operationen für Domänenklassen verzichtet. Somit wurden mit der Transformation von Domänenklassen und (GF-)Anwendungsfällen in entsprechende Spezifikationsklassen bisher lediglich die Struktursicht und die Funktionsicht der Anforderungsspezifikation im Klassenmodell der Softwarespezifikation integriert. Nun muss noch die Verhaltenssicht abgebildet werden.

Dazu wird die in der Anforderungsermittlung eingenommene »Black-Box«-Sicht auf Interaktionen zwischen Akteuren und dem Anwendungssystem (vgl. Abschnitt 20.2) zu einer »Grey-Box«-Sicht auf Interaktionen zwischen Akteuren und Instanzen von Spezifikationsklassen verfeinert, wofür letztere mit Operationen auszustatten sind.

Um das Ablaufverhalten der Anwendungsfälle einheitlich in die Softwarespezifikation abbilden zu können, werden für die drei Stereotype von Spezifikationsklassen («AF-Klasse», «GF-Klasse» und «Entitätsklasse») zunächst die folgenden Standardoperationen vereinbart, die bei der Transformation der Anwendungsfälle automatisch erzeugt werden können.

Jede AF-Klasse erhält eine Operation zur Aktivierung des entsprechenden Anwendungsfalls. Diese wird mit dem Namen des Anwendungsfalls bezeichnet und benötigt ggf. Parameter, mit denen AF-Instanzen z.B. im Falle von include-Beziehungen die notwendigen Informationen austauschen können.

Muss der Akteur im Verlauf des Anwendungsfalls Instanzen z.B. einer Entitätsklasse EK selektieren, wird für die AF-Klasse eine Standardoperation `selektiereEK()` definiert. Ausgaben des Systems an den Akteur können – wie bereits in der Anforderungsspezifikation – als Rückgabewerte modelliert werden. So zeigt z.B. die Angabe `praesentiereEK` auf dem Rückkehrpfad eines Aufrufs von `selektiereEK()`, dass die Attributwerte der selektierten Instanz der Entitätsklasse EK ausgegeben werden. Interessieren in einem bestimmten Kontext nur einige Attributwerte der Instanz, sollte dies mit entsprechenden Aufrufen von `gib..()`-Standardoperationen dargestellt werden.

Die Eingabe bzw. Modifikation von Attributwerten selektierter Objekte durch den Akteur kann abstrakt durch eine nicht weiter spezifizierte `modifiziereEK()`-Standardoperation oder konkret durch eine oder mehrere `setze..()`-Standardoperationen modelliert werden. Die entsprechenden Entitätsklassen werden der Beschreibung des Anwendungsfalls entnommen. Für eine automatische Transformation kann die bei der Verifikation der Anforderungsspezifikation erstellte CRUD-Matrix (vgl. Abschnitt 21.2) herangezogen werden, die angibt, welche (Domänen-)Klassen für welche Anwendungsfälle relevant sind.

Natürlich sind für Entitätsklassen alle allgemeinen Standardoperationen verfügbar (vgl. Tab. 9–1 auf Seite 102). Um diese Operationen auch den Akteuren zur Verfügung zu stellen, können AF-Klassen sie für die von ihnen repräsentierten Entitätsobjekte anbieten und »per Delegation« an die Entitätsobjekte weiterleiten. Treten hierbei Namenskonflikte auf, z.B. wenn eine AF-Klasse Instanzen unterschiedlicher Entitätsklassen mit teilweise gleichbezeichneten Attributen präsentiert, wird in den Bezeichnungen der entsprechenden `gib()`- und `setze()`-Standardoperationen der Name der Entitätsklasse dem Attributnamen vorangestellt.

Die resultierenden Standardoperationen für AF-Klassen sind in Tabelle 24–1 zusammengefasst. Diese Operationen werden auch für GF-Klassen definiert.

Nachricht/Operation	Beschreibung
<code>AWFName()</code> bzw. <code>GP-FName()</code>	Start des Anwendungsfalls mit dem Namen »AWFName« bzw. der GP-Funktion »GP-FName«. Rückgabewert kann z.B. eine einzelne oder eine Menge von Instanzen einer Entitätsklasse sein oder ein Erfolgsindikator.
<code>praesentiereEKNamen</code>	Rückgabewert: Präsentation der Namen aller Instanzen von Entitätsklasse EK.
<code>selektiereEK()</code>	Selektion einer Instanz von Entitätsklasse EK. Rückgabewert können die Attributwerte der vom Akteur selektierten Instanz sein.
<code>praesentiereEK</code>	Rückgabewert: Präsentation der Attributwerte einer zuvor selektierten Instanz von Entitätsklasse EK.
<code>modifiziereEK()</code>	Änderung von Attributwerten einer zuvor selektierten Instanz von Entitätsklasse EK.

**Tab. 24–1** Standardoperationen und Rückgabewerte für AF- und GF-Klassen

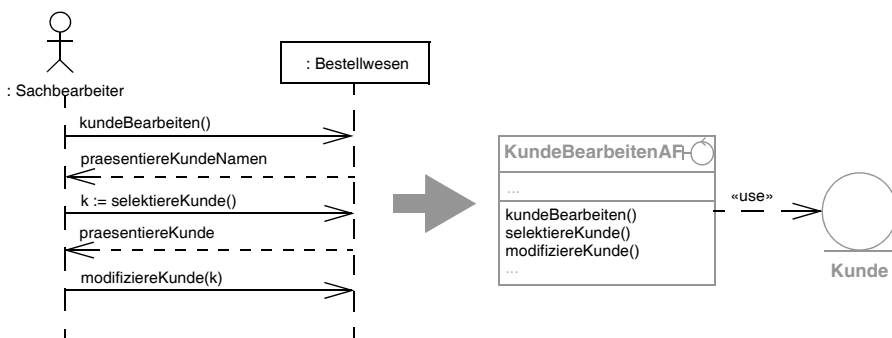


Zur Auskleidung des Klassenmodells der Softwarespezifikation mit weiteren, fachlich motivierten Operationen liefern die in der Anforderungsspezifikation festgehaltenen Anwendungsfälle, Aktivitäts- und ggf. Zustandsdiagramme zusammen mit den Interaktionsdiagrammen (vgl. Abschnitt 20.2) die Basis. Hierzu benötigte Nicht-Standardoperationen werden wie folgt aufgenommen und ihre Verantwortlichkeiten textuell festgehalten.

Ist das Ablaufverhalten eines Anwendungsfalls bzw. einer GP-Funktion durch ein Aktivitätsdiagramm modelliert, wird jede vom System unterstützte Aktivität in eine entsprechend bezeichnete Operation der zugehörigen AF- bzw. GF-Klasse transformiert. Für jedes Interaktionsdiagramm der Anforderungsspezifikation werden die aus Domänenklassen und Anwendungsfällen transformierten Entitäts- und AF-Klassen ermittelt und mit Assoziationen bzw. Abhängigkeiten und Operationen entsprechend der Nachrichten im Interaktionsdiagramm ergänzt. Jede in einem Interaktionsdiagramm für einen Anwendungsfall vom Akteur an das Anwendungssystem gesendete Nachricht wird also auf eine entsprechende Operation der AF-Klasse abgebildet. Ebenso führen (Aufruf-)Ereignisse eines Zustandsdiagramms zu entsprechenden Operationen in derjenigen Entitäts-, AF- bzw. GF-Klasse, für deren Anforderungsspezifikations-Pendant das Zustandsdiagramm gilt.

Textuelle Spezifikationen von Anwendungsfällen und Aktionen werden mittels Vor- und Nachbedingungen der entsprechenden Operationen präzisiert. Damit die Durchgängigkeit und Verfolgbarkeit gewahrt ist, verwendet man für die Operationen die Bezeichner der Aktionen, Nachrichten und Ereignisse aus der Anforderungsspezifikation.

**Beispiel 24–6** Für das in Abbildung 24–8 links gezeigte Interaktionsdiagramm aus der Anforderungsspezifikation des Bestellwesens sind rechts die resultierenden Operationen der Spezifikationsklasse `KundeBearbeitenAF` und die Benutzungsabhängigkeit zur Entitätsklasse `Kunde` dargestellt.



**Abb. 24–8** Aus dem Sequenzdiagramm resultierende USE-Abhängigkeit und Operationen

Mit der Bestimmung der fachlich motivierten Operationen geht gleichzeitig die Präzisierung des Verhaltens der Anwendungsfälle einher. Genauere Ausführungen zur Identi-

fizierung solcher Operationen folgen daher in Kapitel 26. Wichtig ist, sich immer wieder klar zu machen, dass die Auskleidung des Klassenmodells der Softwarespezifikation mit Operationen keine Realisierung darstellt, sondern lediglich die mit umgangssprachlichen Vor- und Nachbedingungen formulierten deskriptiven Anwendungsfälle präzisiert und in präskriptiver Form »operationalisiert«. Dies verankert die bislang rein »syntaktisch« transformierten Anwendungsfälle auch »semantisch« im Klassenmodell und ist eine wesentliche Voraussetzung für den in der Verifikation durchzuführenden wichtigen Abgleich der verschiedenen Modelle gegeneinander (vgl. Kapitel 27). Die präzise Modellierung des Ablaufverhaltens der fachlichen Operationen ist Gegenstand des Entwurfs.