

Die Teil – Ganzes – Beziehungen

In diesem Text geht es um die beiden Varianten der Teil-Ganzes-Beziehung, also um die **Aggregation** und die **Komposition**. Beide stellen Spezialfälle der Assoziation dar, unterscheiden sich von der einfachen Assoziation also dadurch, dass an einem Ende der Beziehung Teile eines Objektes von anderen Ende der Beziehung stehen. Die typische sprachliche Formulierung dieses Typs von Beziehung ist "... **besteht aus** ...".

Beispiele

Im Raumplaner-Projekt ist die Beziehung der Klasse Moebel zur Klasse Zeichenflaeche eine reine **Assoziation**. Die Moebel-Objekte benutzen das Leinwand-Objekt zu ihrer Darstellung, sind aber nicht Teil des Leinwand-Objektes. Damit hat auch zu tun, dass auf dem Leinwand-Objekt nicht Moebel-Objekte gezeichnet werden, weswegen das Leinwand-Objekt von den Moebel-Objekten die Methoden GibFigur() und GibFarbe() aufrufen muss, da sie nicht die Objekte selbst, sondern deren grafische Repräsentation verwendet. Zwischen dem Grafikfenster-Objekt und dem Leinwand-Objekt dagegen besteht eine Teil-Ganzes-Beziehung, da das Panel, also das Inhaltsfenster ein Teil des Grafikfenster-Objektes ist.

Ein weiteres Beispiel ist die Bildschirmoberfläche, auf der Icons zu finden sind. Sie sind sicherlich Teile der Oberfläche.

In diesem Text soll es nun darum gehen, zwischen den beiden Fällen Aggregation und Komposition zu unterscheiden. Welcher Fall ist es denn nun?

Aggregation

Dazu schreibt Oesterreich [s.u.]:

"Unter einer Aggregation versteht man die Zusammensetzung eines Objektes aus einer Menge von Einzelteilen. ... Kennzeichnend für alle Aggregationen ist, daß das Ganze Aufgaben stellvertretend für seine Teile wahrnimmt. Die Aggregatklasse enthält beispielsweise Operationen, die keine unmittelbare Veränderung im Aggregat selbst bewirken, sondern die Nachricht an seine Einzelteile weiterleiten."

Beispiel

Im Raumplaner-Projekt besteht diese Beziehung beispielsweise zwischen der Schrankwand und den Schrank-Objekten aus denen sie besteht. Ein Beispiel für das Weiterreichen der Anforderungen ist das Zusammensetzen des path-Objektes aus den einzelnen path-Objekten¹.

Darstellung

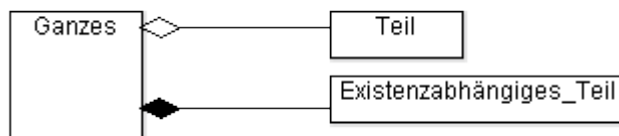
Eine Aggregation wird wie eine Assoziation als Linie zwischen zwei Klassen dargestellt. Der Unterschied der Darstellung zur einfachen Assoziation ist eine kleine Raute an der Klassendarstellung des Ganzen, also dort, von wo aus man die Formulierung "... besteht aus ..." ansetzt.

Komposition

"In manchen Fällen beschreiben Aggregationen Beziehungen, in denen die Teile vom Ganzen existenzabhängig sind. Das heißt, wenn das Aggregat (das Ganze) gelöscht wird, werden alle Einzelteile ebenfalls gelöscht. Wird ein Einzelteil gelöscht, bleibt das Aggregat erhalten."

Darstellung

Statt der einfachen offenen Raute wird bei der Komposition eine ausgefüllte Raute verwendet. Das Bild rechts zeigt die unterschiedlichen Darstellungen.



Kardinalitäten

Die UML kennt noch die Darstellung der unterschiedlichen Kardinalitäten an den Verbindungslinien, also der Frage, wie viele Objekte auf jeder Seite beteiligt sein können. In den Betrachtungen hier wird darauf nicht eingegangen. Allerdings lässt sich für Kompositionen eindeutig fordern, dass es beim Ganzen hier eine 1 sein muss.

Weitere Texte zu Aggregation und Komposition

Die auf den vorigen Seiten angegebenen Zitate stammen aus dem sehr empfehlenswerten Buch¹ von Bernd Oestereich zu OO Softwareentwicklung "Analyse und Design mit der UML". Hier noch eine Darstellung aus dem aktuelleren Buch "UML 2 glasklar"²:

"Aggregation

Eine Aggregation (im Beispiel die Personengruppe) drückt eine „Teile-Ganzes-Beziehung“ aus. Die aggregierten Instanzen einer Klasse sind dabei Teil eines Ganzen, das durch die Klasse am anderen Beziehungsende repräsentiert wird.

Die UML-Spezifikation legt die Semantik einer Aggregation auf die Lesart „besteht aus“ fest. Demnach ist die Aggregation zu lesen als „Personengruppe besteht aus Menschen“. Dasjenige Assoziationsende, welches mit dem leeren kleinen Diamanten versehen ist, wird als „Ganzes“ aufgefasst, seine „Teile“ befinden sich als Klasse am anderen Assoziationsende.

Im Kern ist die Aggregationsbeziehung nichts anderes als eine abkürzende Schreibweise der Rollen „besteht aus“ - für das Assoziationsende des Ganzen - und „ist Teil von“ - für das Ende der Teile.

Weitere Einschränkungen trifft die Modellierungssprache an dieser Stelle nicht. Insbesondere beschränkt sie nicht ausdrücklich, zu wie vielen verschiedenen „Ganzen“ ein „Teil“ gleichzeitig beitragen kann ...

Komposition

Eine strengere Form des Zusammenhanges wird durch die Komposition definiert. Sie drückt im Gegensatz zur Aggregation die physische Inklusion der Teile im Ganzen aus. Teile und Ganzes bilden eine Einheit, deren Auflösung durchaus die Zerstörung des Ganzen zur Folge haben kann. Im Beispiel besitzt der Mensch 10 Finger, die zusammen eine Einheit bilden.

Deshalb gilt hier die verschärfende Einschränkung, dass ein Teil zu einem Zeitpunkt höchstens genau einem Ganzen zugeordnet sein darf. Andernfalls würde es zu wechselseitigen Abhängigkeiten zwischen verschiedenen Ganzen kommen, die die Zerstörung verschiedener Ganzer als Folge der Zerstörung eines gemeinsamen Teiles bedeuten könnten, (Ein Finger ist genau einem Menschen zugeordnet.)"

1 Bernd Oestereich: Objekt-orientierte Softwareentwicklung; Analyse und Design mit der UML; Oldenbourg – Verlag. Zur Zeit gibt es davon die achte aktualisierte Ausgabe im Oldenbourg – Verlag.

2 Rupp, Hahn, Queins, Jeckle, Zengler: UML 2 glasklar; Praxiswissen für die UML-Modellierung und -Zertifizierung; Hanser - Verlag

Was ist es denn nun?

Für die Schrankwand muss man nach diesen Formulierungen von einer Komposition ausgehen, da jedes Schrank-Objekt genau einem Schrankwand-Objekt zuzuordnen ist.



Sind aber Schrank-Objekte jeweils für sich "lebensfähig" im Sinne der Modellierung, weil sie auch ohne die Zugehörigkeit zu einem Schrankwand-Objekt existieren können, handelt es sich um eine Aggregation¹. Daher ist in diesem Beispiel der konkrete Anwendungsfall entscheidend.

Assoziation, Aggregation und Komposition sind Objektbeziehungen

Ein ganz wichtiger Aspekt taucht in diesem Text mehrfach auf:

Diese Beziehungen sind **Objektbeziehungen** und keine Klassenbeziehungen, obwohl man sie im Klassendiagramm darstellen kann.

Die Vererbung ist eine Beziehung zwischen Klassen, ein Schrankwand-Objekt kann aber immer nur als konkreten Schrank-Objekten bestehen.

Das hat Auswirkungen auf das Auftreten im Programmtext: Häufig wird man den Aufruf des Konstruktors der Klasse der Teile im Programmtext der Klasse des Ganzen vorfinden. Zumindest muss das Ganze eine Referenz auf die Teile-Objekte haben.

Bei der Vererbung wird zwar auch der Konstruktor der vererbenden Klasse aufgerufen, das ist aber Teil des Initialisierungsprozesses des einen Objektes, was man bei Python an self als erstem Parameter erkennen kann.

¹ Oder sollte es vielleicht doch eher darauf ankommen, dass im konkreten Fall eines Schrankwand-Objektes die Schrank-Objekte, die ihm zugeordnet sind, nur ihm zugeordnet sein können, egal, was mit einzelnen Schrank-Objekten ist?

Die Schwierigkeit, diese Frage eindeutig zu klären, sollte deutlich machen, dass man im Unterricht den Schülerinnen und Schüler die beiden Fälle vermitteln kann und es interessant sein kann, mit ihnen die Zuordnung zu diskutieren, eine abgesicherte Entscheidung aber nicht Prüfungsstoff sein kann.

In einem konkreten Projekt ist zudem allein die Frage interessant, an welchen Stellen der jeweils andere Typ zu konkreten Anforderungen im Projekt führt.