Analyseklassendiagramm

# Einleitung

Das Analyseklassendiagramm stellt den ersten Entwurf der Klassen dar, die man aus dem Lastenheft identifizieren kann. Hierbei geht es vor allem um die Klassen, die Entities abbilden und nicht um solche, die nur um des Programms willen existieren (wie z.B. Datenbankmanager, etc.).

# Hinweis

Da bei der Modellierung nicht die Personenverwaltung berücksichtigt werden muss, wurde der Systemnutzer nicht modelliert.

# Pattern und HIWEISE?

Für das Diagramm werden das Rollen- und das Koordinatorpattern verwendet. Da einige Klassen mehr als eine Referenz auf eine Klasse haben, bietet sich hier dieses Pattern an. Das Koordinatorpattern wird nur einmalverwendet, mehr dazu an der passenden Stelle.

Auf die Labels der Relationen wurde aus Übersichtszwecken verzichtet. Außerdem werden die Attribute die einer Rolle zugewiesen sind immer auch als Attribut vermerkt, d.h. die Rollen ersetzten keine Attribute

# Entities

## Hochzeitsveranstaltung

Eine Hochzeitsveranstaltung hat einen Titel und ein Motto. Um die verschiedenen Rollen, die eine Person innerhalb einer Hochzeitsveranstaltung innehaben kann abzubilden, benutzt sie hier das Rollen-Pattern. Es gibt zwei Personen, die das Brautpaar abbilden, 1 Person als Berater, 0..\* als Gäste und als Unterhaltungsmanager

## Person

Eine Person hat einen Namen, eine Liste von Email Adressen und eine Liste von Telefonnummern, beiden vom Typ String. Für die Telefonnummern wurde der Typ String gewählt da man Telefonnummern eher selten als wirkliche Zahlen betrachtet. Außerdem in einer 1..1 Relation der Adresse ein Ort zugewiesen. Als letztes kennzeichnet ein boolsches Flag, ob es sich bei der Person um einen Dienstleister handelt.

## Aktion

Eine Aktion hat einen Titel und eine Beschreibung. Außerdem ist die Aktion einer Hochzeit per Komposition zugeordnet, denn es macht keinen Sinn eine Aktion zu behalten, wenn die dazugehörige Hochzeit nicht mehr existiert. Des Weiteren besitzt eine die Aktion eine Art, einen Zustand und eine Notiz. Diese alle sind vom Typ String. Der Anfang und das Ende der Aktion werden als DateTime realisiert. Die Teilnehmerliste ist eine 0..\* Relation zu den Personen, die Ortsliste eine 0..\* zu den Orten. Es gibt auch eine Liste an Medien und Belegen die mit 0..\* mit den Medien bzw. Belegen verbunden ist. Die Hilfsmittel sind mit den Aktionshilfsmitteln über eine Komposition verbunden. Das versteckt und das Meilenstein Attribute sind vom Typ boolean. Außerdem gibt es noch eine 1..\* Relation zu den Personen, die die Verantwortlichen beschreibt. Als letztes regelt ein int die Priorität.

## Caterer

Der Caterer besitzt einen Namen und Beschreibung. Außerdem besitzt er 0 bis beliebig viele Belege. Um die Generalisierung des Essens und des Trinkens wieder zu spezialisieren hat er ja ein Attribut für das Essen und das Trinken vom Typ Lebensmittel mit der Kardinalität 0..\*. Hier greift das Rollen-Pattern. Des Weiteren hat er eine (1) Kontaktperson vom Typ Person. Ein boolsches Flag namens zumVergleich regelt, ob dies ein Caterer ist, der liefert oder der nur ein Vergleichsangebot repräsentiert.

## Ort

Da sich bei der Analyse ergeben hat, dass zu einer Adresse viele Zusatzattribute gehören, wurde dies als einzelne Klasse ausgelagert. Ein Ort als somit eine Straße, eine Hausnummer und für besondere Fälle einen Adresszusatz. Alles ist vom Typ String. Des Weiteren gehören zu einem Ort eine Stadt und eine Postleitzahl. Da es in manche Länder üblich ist die Provinz/Land/Bundesland mit abzuspeichern, gibt es das Attribut Provinz. Als letzes wird das Land abgespeichert.

## Lebensmittel

Bei der Analyse wurde festgestellt, dass die Attribute von Essen und Trinken gleich sind. Außerdem sind sie in der realen Welt daraus einer Überkategorie zuteilbar. Daher wurden das Essen und das Trinken in einer Entität, das „Lebensmittel“ vereinigt.

Es hat einen Titel und eine Beschreibung, eine Menge und eine Mengenbeschreibung. Die Menge ist hierbei als double zu sehen, da hier verschiedenste Werte, also auch Gleitkommazahlen stehen können. Um nun zu wissen was die Zahl ausdrücken will, also z.B. Kilogramm, Liter, Gläser, Flaschen existiert geradeeben die Mengenbeschreibung.

Lebensmittel kann in der Kardinalität 0..\* beim Caterer auftreten. Dies wurde aus dem Lastenheft entnommen, da ein Lebensmittel durchaus ohne Caterer existieren kann, aber auch aufgrund der Wiederverwendbarkeit auch mehreren zugewiesen sein kann.

## Beleg

Ein Beleg stellt eine Rechnung oder eine Quittung der realen Welt dar. Um diesen zu identifizieren hat er einen Titel und eine Beschreibung. Für die Kostenkontrolle speichert der die Kosten inkl. der Währung. Außerdem referenziert er beliebig viele aber mindestens ein (1..\*) Medium.

## Hilfsmittel

Ein Hilfsmittel hat ebenfalls einen Titel und eine Beschreibung, beide vom Typ String. Außerdem ist ein Hilfsmittel von einer Art. Diese ist ebenfalls ein String. Beliebig viele (also 0..\*) Belege lassen sich dem Hilfsmittel zuordnen.

## Medium

Ein Medium hat einen Titel, einen Typ und eine URI, die den Pfad zu der repräsentierten Datei darstellt. Alle sind vom Typ String.

## Aktionshilfsmittel

Diese Klasse ist nach dem Koordinator-Pattern entstanden. Da eine Aktion mehrere Hilfsmittel mit einem Attribut nämlich der Menge speichern soll, muss dies in eine separate Klasse ausgegliedert werden. Das Aktionshilfsmittel ha somit eine Referenz auf die Aktion und das Hilfsmittel gespeichert. Außerdem die Menge des jeweiligen Hilfsmittels, welches die Aktion verwendet. Somit ist eine Komposition zu der Aktion als auch zu dem Hilfsmittel gegeben, denn wenn eines der beiden Entitäten nicht mehr existiert, mach die Relation keinen Sinn mehr. Somit ist die Instanz der Klasse existenzabhängig von den beiden Entitäten.