

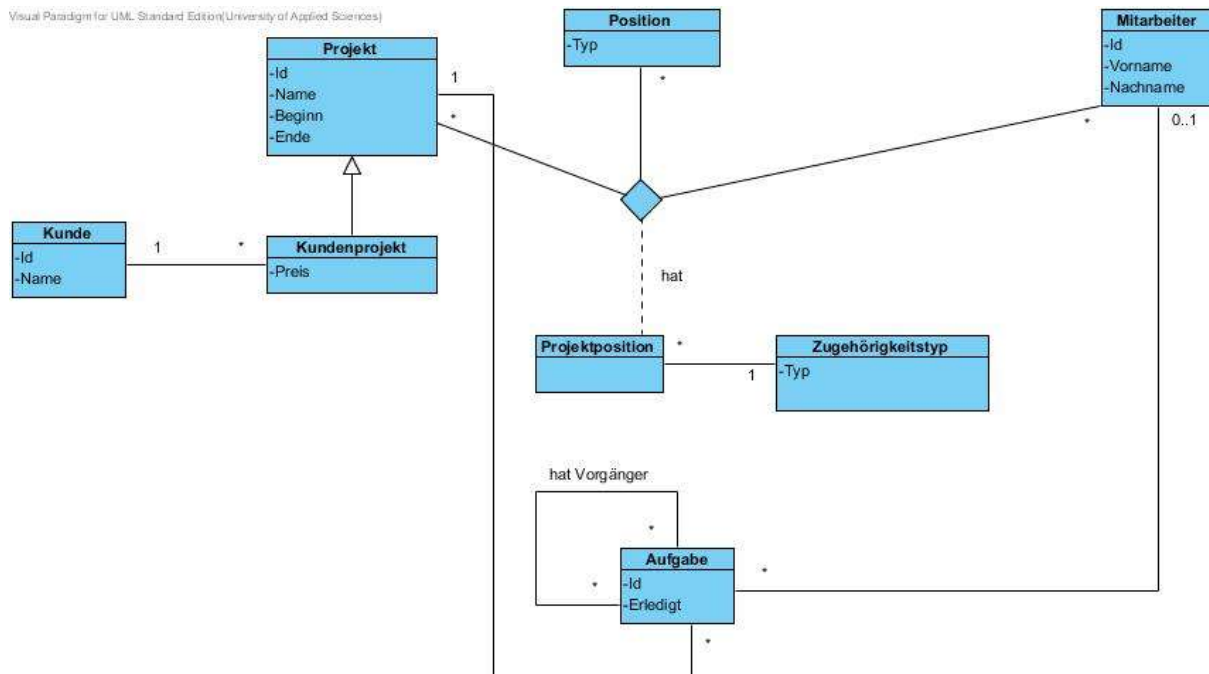
1	Projektmanagement	2
1.1	UML.....	2
1.2	Relationales Model (textuell)	3
2	Terminplanung	4
2.1	UML.....	4
2.2	Relationales Model (textuell)	5
3	Spedition	6
3.1	Teilaufgabe A	6
3.2	Teilaufgabe B	6

1 Projektmanagement

Folgend ist die Dokumentation für die Aufgabe Projektmanagement angeführt.

1.1 UML

Folgend ist das UML Diagramm für diese Aufgabe angeführt.



Annahmen:

Da ein Projekt entweder ein internes oder externes Projekt sein kann, aber ein internes Projekt keine zusätzlichen Attribute benötigt ist hier keine eigene abgeleitete Klasse nötig.

Da eine Aufgabenanalyse erst nach dem Anlegen eines Projekts durchgeführt wird ist eine 1 - * Beziehung zwischen Projekt und den Aufgaben definiert, da erst nach der Analyse die Aufgaben feststehen und angelegt werden.

Auch bei der Beziehung zwischen den Benutzern und Aufgaben wird eine 0..1 - * Beziehung definiert, da die Aufgaben angelegt sind bevor eine Beziehung zu den Benutzern hergestellt wird.

Die * - * Beziehung zwischen den Aufgaben definiert dass eine Aufgabe nicht zwingend einen Vorgänger bzw. einen Nachfolger haben muss.

Das Flag erledigt ist nicht auf der n:m Beziehung zwischen den Aufgaben, da eine Aufgabe für sich alleine stehen kann und daher hier das Erledigt Flag benötigt wird.

1.2 Relationales Model (textuell)

Folgend ist das textuelle Relationen Modell angeführt.

Sollten Attribute nullable sein so wird dies explizit angegeben, ansonsten sind not null Restriktionen implizit anzunehmen.

Kunde (Id (**PK**), Name)

Projekt (Id (**PK**), Name, Beginn, Ende)

Kundeprojekt (Projekt_Id (**PK - FK - Projekt**), Preis, Kunde_Id (**FK - Kunde**))

Position (Typ (**PK**))

Zugehörigkeitstyp (Typ (**PK**))

Mitarbeiter (Id (**PK**), Vorname, Nachname)

Projektposition (Projekt_Id (**PK - FK - Projekt**), Position_Id (**PK - FK - Position**),

Mitarbeiter_Id (**PK - FK - Mitarbeiter**), Zugehörigkeitstyp_Id (**FK - Zugehörigkeitstyp**))

Aufgabe (Id (**PK**), Erledigt, Mitarbeiter_Id (**nullable FK - Mitarbeiter**), Projekt_Id (**FK - Projekt**))

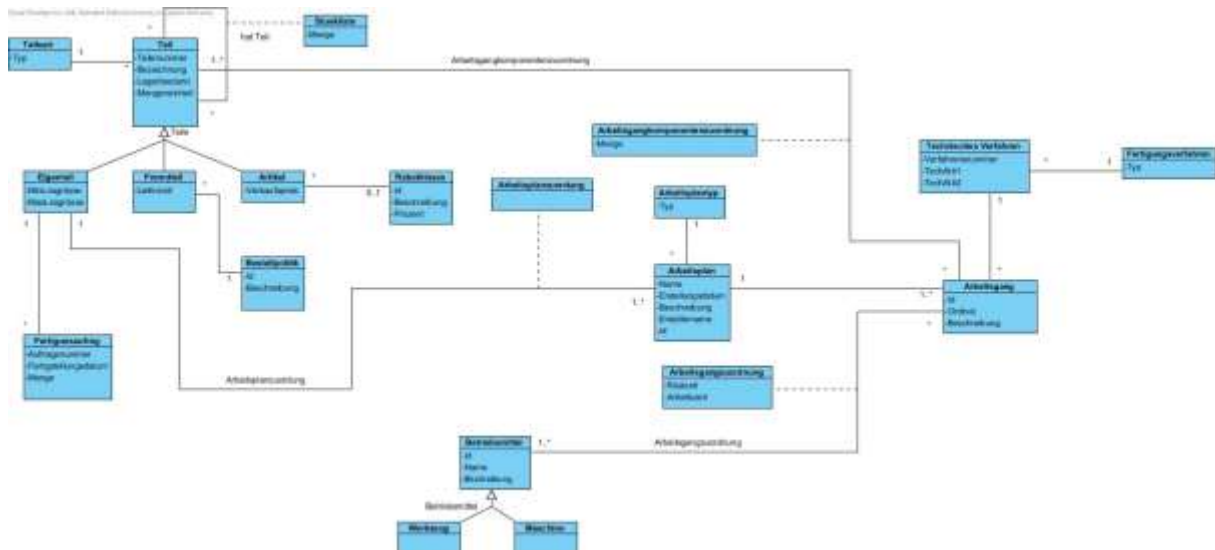
Aufgabenzuweisung (Vorgänger_Aufgabe_Id (**PK – FK - Aufgabe**), Nachfolger_Aufgabe_Id (**PK – FK - Aufgabe**))

2 Terminplanung

Folgend ist die Dokumentation für die Aufgabe Terminplanung angeführt.

2.1 UML

Folgend ist das UML Diagramm für diese Aufgabe angeführt.



Annahmen:

Da nicht genau spezifiziert gehe ich davon aus das als Unterteil jede Art von Teil verwendet werden kann. Es ist zwar angemerkt das Fremdteile und Artikel als Unterteile verwendet werden können jedoch ist auch angemerkt das ein Teil entweder Oberteil oder Unteil ist, daher die Entscheidung das auch Eigenteile als Unterteil agieren können.

Jedes Fremdteil muss einer Bestellpolitik zugewiesen sein.

Ein Artikel ohne eine Rabatklasse ist ein Artikel der ohne Rabat verkauft wird.

Da keine Attribute für das Technische Verfahren angegeben wurden werden diese generisch genannt (Bsp.: TechAttr1,...)

Es wurden Typisierungsentitäten eingefügt, da diese Typen eindeutig und global innerhalb der Datenbank dargestellt werden sollen.

Rabatkategorie, Bestellpolitik, Arbeitsplanart, Teilart, Fertigungsverfahren

2.2 Relationales Model (textuell)

Folgend ist das textuelle Relationen Modell angeführt.

Sollten Attribute nullable sein so wird dies explizit angegeben, ansonsten sind not null Restriktionen implizit anzunehmen.

Teileart (Typ **(PK)**)

Rabatkasse (Id **(PK)**, Beschreibung, Prozent)

Bestellpolitik (Id **(PK)**, Beschreibung)

Teile (Teilenummer **(PK)**, Bezeichnung, Lagerbestand, Mengeneinheit, Teileart_Id **(FK - Teileart)**)

Stueckliste (Oberteil_Id **(PK – FK - Teil)**, Unterteil_Id **(PK – FK - Teil)**, Menge)

Eigenteil (Teil_Id **(PK – FK - Teil)**, MinLosgrösse, MaxLosgrösse)

Fremdteil (Teil_Id **(PK – FK - Teil)**, Lieferzeit, Bestellpolitik_Id **(FK - Bestellpolitik)**)

Artikel (Teil_Id **(PK – FK - Teil)**, Rabatkasse_Id **(nullable FK - Rabatkasse)**)

Fertigungsverfahren (Typ **(PK)**)

Arbeitsplantyp (Typ **(PK)**)

Fertigungsauftrag (Auftragsnummer **(PK)**, Fertigstellungsdatum, Menge, Teil_Id **(FK - Teil)**)

Technisches Verfahren (Verfahrensnummer **(PK)**, TechAttr1, TechAttr2, Fertigungsverfahren_Id **(FK - Fertigungsverfahren)**)

Arbeitsgang (Id **(PK)**, Ordinal, Beschreibung, Arbeitsplan_Id **(FK - Arbeitsplan)**,

Technisches_Verfahren_Id **(FK – Technisches Verfahren)**)

Arbeitsplan (Id **(PK)**, Name, Beschreibung, Erstellername, Erstellungsdatum, Arbeitsplantyp_Id **(FK - Arbeitsplantyp)**)

Arbeitsgangkomponentenzuordnung (Teil_Id **(PK – FK - Teil)**, Arbeitsgang_Id **(PK – FK - Teil)** , Menge)

Arbeitsplanzuordnung (Eigenteil_Id **(PK – FK - Eigenteil)**, Arbeitsplan_Id **(PK – FK - Arbeitsplan)**)

Betriebsmittel (Id **(PK)**, Name, Beschreibung)

Werkzeug (Betriebsmittel_Id **(PK – FK - Betriebsmittel)**)

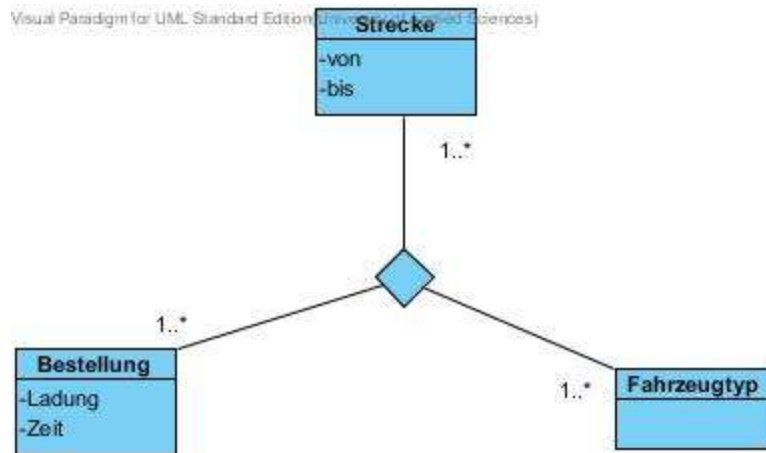
Maschine (Betriebsmittel_Id **(PK – FK - Betriebsmittel)**)

Arbeitsgangzuordnung (Betriebsmittel_Id **(PK – FK - Betriebsmittel)**, Arbeitsgang_Id **(PK – FK - Betriebsmittel)**, Rüstzeit, Beschreibung)

3 Spedition

Folgend ist die Dokumentation für die Aufgabe Spedition angeführt.

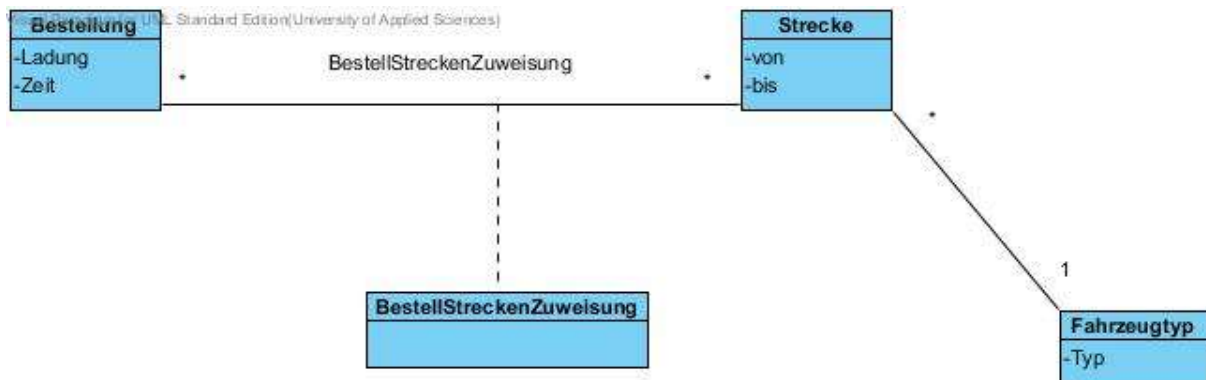
3.1 Teilaufgabe A



1..* Beziehung zwischen den ternär verknüpften Entitäten, da:

1. Eine Bestellung von mehreren Fahrzeugen über mehrere Strecken transportiert werden kann
2. Über eine Strecke mehrere Bestellungen mit verschiedenen Fahrzeugen transportiert werden kann
3. Ein Fahrzeugtyp auf mehreren Strecken verwendet werden darf und auch mehrere Bestellungen über diese transportiert.

3.2 Teilaufgabe B



Eine ternäre Beziehung ist hier nicht mehr notwendig da ein Fahrzeugtyp genau einer Strecke zugeordnet ist und daher keine Mehrfachverwendungen mehr abzubilden sind. Der Fahrzeugtyp kann hierbei über die Strecke identifiziert werden.