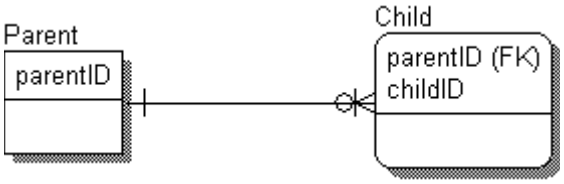
Explizit Association (Explizite Beziehung)

Absicht

Zwischen zwei Elementen sollen Beziehungen erstellt werden, die eine unterschiedliche Bedeutung haben.

Problem



Durch eine Assotiation wird eine Beziehung zwischen zwei Elementen modelliert. In viele Modellierungsprachen wie UML sind Beziehungen ein wesentlicher Bestandteil der Beschreibung von Modellen.

Association in UML: **Association** represents the ability of one instance to send a message to another instance. This is typically implemented with a pointer or reference instance variable, although it might also be implemented as a method argument, or the creation of a local variable.

Association in ER-Modelling: Eine Beziehung stellt eine Verknüpfung zwischen einer oder zwei Entitäten dar.

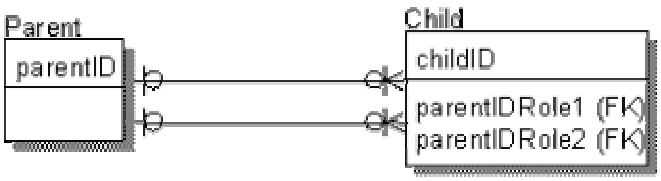
Eine Beziehung resultiert immer aus der fachlichen Notwendigkeit, Entitäten zusammen aus funktionaler Sicht zu betrachtet. Der Informationsgehalt besteht aus einenm strukturellen und einem semantischen Charakter.

Der strukturelle Charakter eine Beziehung wird ausgedrückt durch

- die identifizierten Elemente
- Kardinalität
- ob es sich um eine Muss oder Kann-Beziehung handelt.
- ob es sich um eine identifizierende oder um eine nicht identifizierende Beziehung handelt

Der semantische Charakter einer Beziehung wird beschrieben durch die fachliche Notwendigkeit.

Manchmal bestehen zwischen zwei Elemenen mehr als eine semantische Beziehung. Im Datenmodell wird dieser Zustand oft durch mehrere Beziehungen ausgedrückt.



Die Abbildung unterschiedlicher semantischer Beziehungen durch Assozationen macht Sinn, solange

1. die Anzahl der semantischen Beziehungen begrenzt ist (< 4)
2. Anzahl und Kontext der Beziehung statisch ist und sich im Anwendungkontext nicht ändert.

Trifft dies in einem fachlichen Zusammenhang nicht zu, müssen andere Methoden verwendet werden um die Beziehung zwischen den Elemente zu beschreiben.

Forces

- Zwischen zwei Entites bestehen fachliche Beziehungen. Die Anzahl der Beziehungen ändert sich im Anwendungskontext bzw. könnte sich ändern, so dass das physiklische DM geändert werden muss.

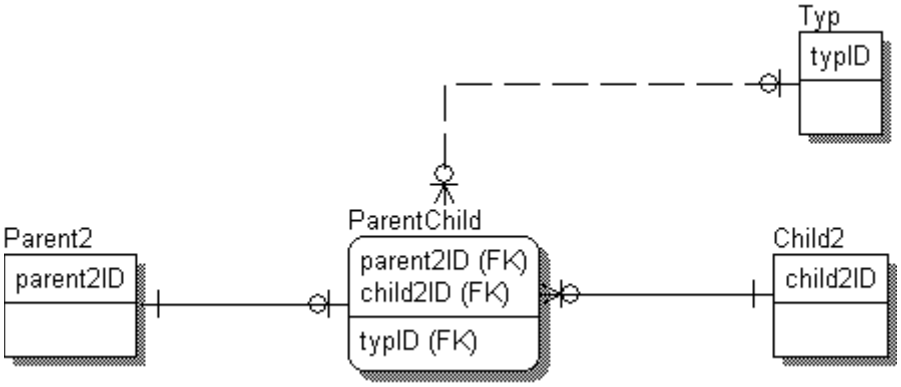
Lösung

Zur Lösung des Problems erstellen wir eine neue Entität, die anstelle der urprünglichen Beziehung diese sozusagen erstetzt. Die neue Entität wird mit der ursprünglichen Entität wieder durch Relationen verbunden.

Durch eine weitere neue Entität (im Beispiel Typ) wird der semantische Charakter der Beziehung typisiert. Somit entsteht zwischen Parent und Child eine Beziehung, deren semantischer Charakter durch den Beziehungstyp näher beschrieben wird.

Die Anzahl der Beziehungen zwischen Parent und Child hängt davon ab, welche Spalten als identifizerende Attribute in der Entität ParentChild deklariert werden. (Siehe dazu Strategien)

Die Anzahl der Einträge in der Tabelle Typ entspricht der Anzahl der Assozitaionen aus dem urspünglichen Modell (Modell vor der Änderung). Einträge in der ParentChild Entität entsprechen einzelnen Instanzen der Assoziation.

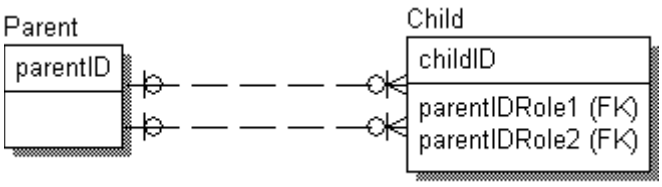


Das wesentliche bei diesem Pattern ist, dass bei der Umwandlung Strukturinformation zum Inhalt des Modells verwandelt wird.

Struktur

Strategien

Anzahl erlaubter Beziehungen

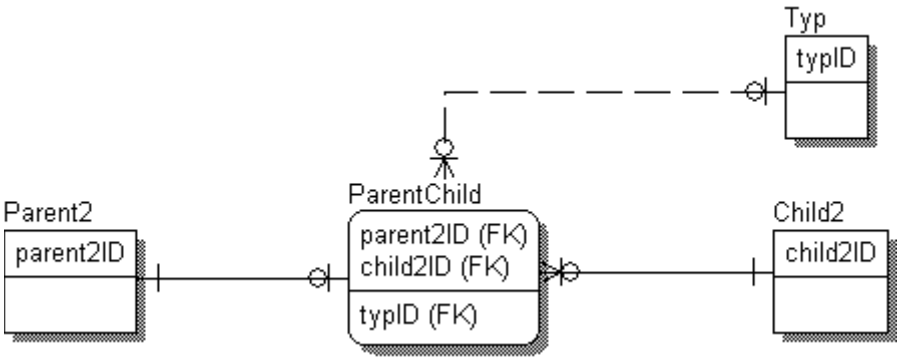


Das obere Bild zeigt noch einmal die Ausgangssituation vor der Anwendung des Patterns. Das Modell erlaubt es, dass der selbe Parent sowohl Rolle 1 als auch Rolle 2 annehmen kann, in dem bei beiden Foreign Keys der selbe Eintrag in der Tabelle erstellt wird.

Dies kann im fachlichen Zusammenhang durchaus richtg sein. Das Datenmodell erlaubt jedoch in dieser Form keine andere Interpretation.

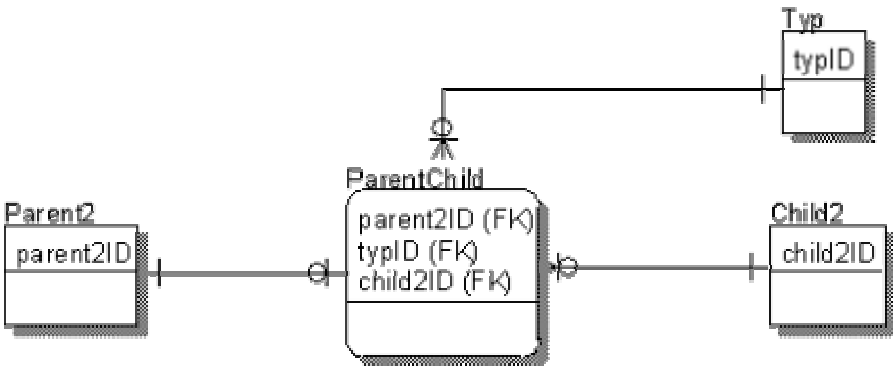
Durch die Anwendung des Patterns kann durch das Design der ParentChild Entität die maximale Anzahl der Beziehungen, d.h. die Anzahl der semantischen Beziehungen, festgelegt werden.

Es existiert nur eine Beziehung zwischen Parent und Child



Wenn die Beziehung zwischen Typ und ParentChild als non-identifying Relationsship modelliert wird, kann ein Parent nur eine Rolle annehmen, weil zwischen Parent und Child nur eine Eintrag möglich ist. Auch dies kann im fachlichen zusammenhang richtig sein. Es sei jedoch hier angemerkt, dass unser Ausgangsmodell mehr als eine Beziehung zwischen Parent und Child zulässt. Dieses Modell ist also nur eine Lösung, wenn zwar zwischen Parent und Child zwar unterschiedliche Beziehungen existieren können, jedoch nur eine Beziehung zu einem Zeitpunkt.

Mehrere Beziehungen unterschiedlichen Typs



Wird die Beziehung zwischen Typ und ParentChild hingegen als identifying Relationsship modelliert wird, sind mehrere Beziehungen zwischen Parent und Child möglich, so dass ein Parent auch mehrere Rollen gleichzeitig annehmen kann. Dies entspricht Modell der Ausgangssituation.

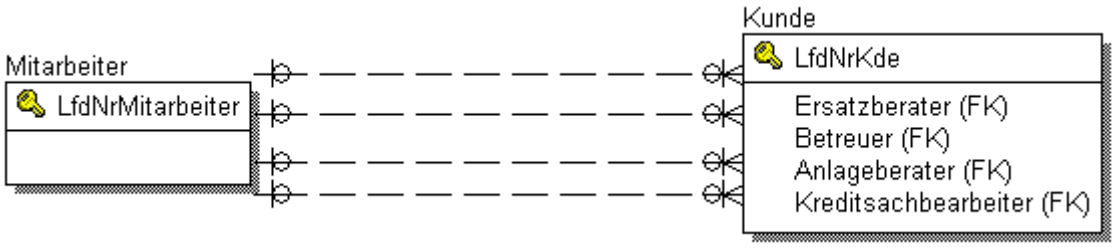
Mehrere Beziehungen, auch unterschiedlichen Typs

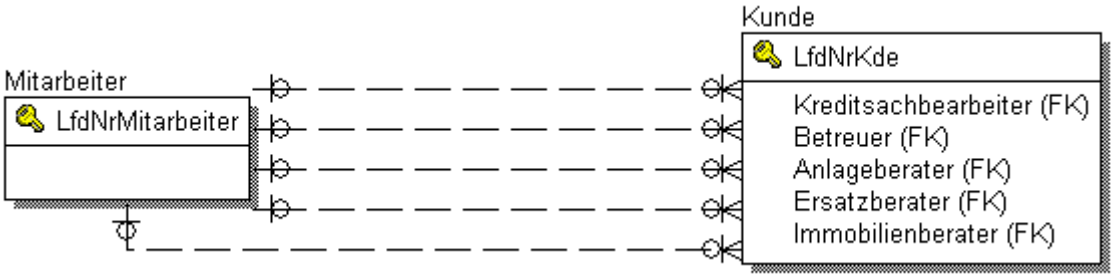
Zusätzlich gibt es noch die Möglichkeit außer dem Feld typeID ein weiters Feld z.B. lfdNr als identifizierendes Attribut zu definieren, bzw. einen künstlichen Schlüssel für die Entität ParentChild zu definieren. In beiden Fällen wäre es dann möglich beliebig viele Beziehungen zwischen Parent und Child – auch gleichen Typs - zu erstellen.

Dies könnte beispielsweise für eine Projetorganisation zutreffen, die zur Definition der Aufbauorganisation von internen Projekten verwendet wird. Eine Beziehung zwischen Mitarbeiter und Projektleiter kann mehfach vorkommen, wenn Mitarbeiter und Projektleiter in mehreren gleichzeitig ablaufenden oder auch aufeinander folgenden Projekten zusammenarbeiten.

Beispiele

Bankkundenbetreuer

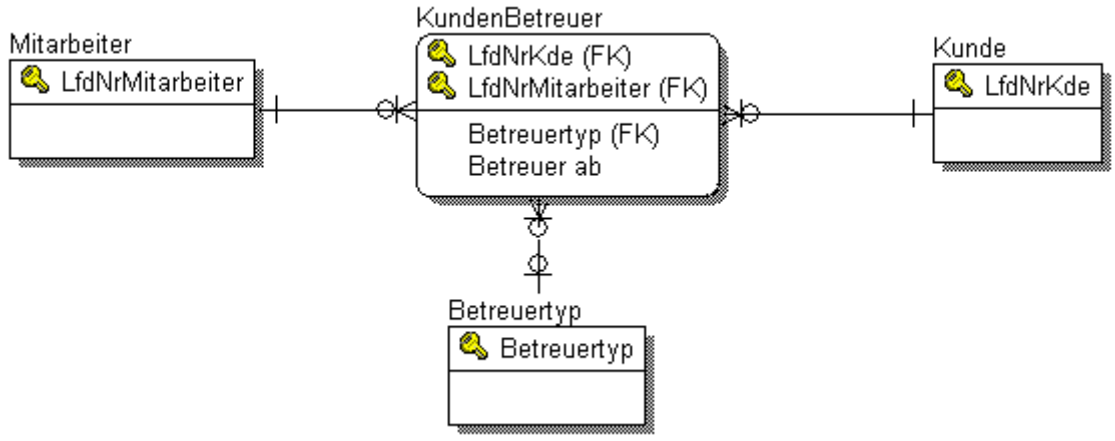




Das obere Bild zeigt die Ausgangssituation für dieses Beispiel. In einer Bank werden dem Kunden Mitarbeiter nach unterschiedlichen Kriterien zugeordnet. Dabei handelt es sich immer um spezielle Rollen, die er Mitarbeiter für den Kunden einnimmt. Das kann ein der Betreuer sein, der als Ansprechpartner für alle möglichen Fragen dient. Das kann ein Anlageberater oder ein Kreditsachbearbeiter sein. Einem Kunden können verschiedenen Mitarbeiter der Bank zugeordnet werden, die in eine dieser Rollen schlüpfen. Dabei können zwei oder mehrere Rollen auch vom selben Mitarbeiter wahrgenommen werden.

Die Beziehung zwischen dem Mitarbeiter und dem Kunden ist in diesem Modell statisch. Die verschiedenen Rollen, die ein Mitarbeiter für den Kunden ausüben kann, werden durch Assoziationen zwischen den beiden Entitäten beschrieben. Werden neue Rollen definiert ist jeweils eine Änderung des Datenmodells notwendig. Bild XXXXX zeigt das Modell nachdem eine neue Rolle *Immobilienberater* hinzugefügt wurde. Besser wäre an dieser Stelle, wenn die Beziehungen, d.h. sowohl Art als auch Anzahl, als Daten beschrieben werden könnte.

Das ExplizitAssoziation Pattern soll nun für das Modell angewandt werden.



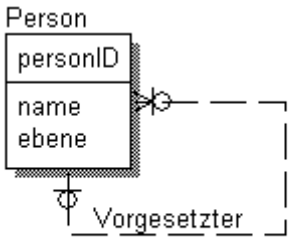
Dazu wird eine Entität KundenBetreuer angelegt und eine Entität Betreueretyp. Die Enität Betreuertyp enthält für jede Beziehung aus dem urprünglichen Modell einen Eintrag. Für das Beispiel würde diese bedeuten, die Tabelle Betreuertyp hätte folgende Einträge

1. Beteuer
2. Ersatzberater
3. Immobilienberater
4. Kreditsachbearbeiter
5. Anlageberater.

Durch die Anwendung dieses Patterns können nun beliebige Rollen von Berateren definiert und dem Kunden zugeordnet werden. Eine Änderung am Datenmodell ist nicht erforderlich.

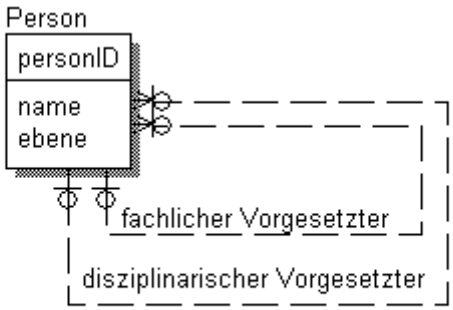
Organisationsstruktur

Dieses Beispiel zeigt, dass sich das Pattern auch auf hierarchische Beziehungen auf sich selbst angewandt werden können. Parent und Child sind dabei identisch.

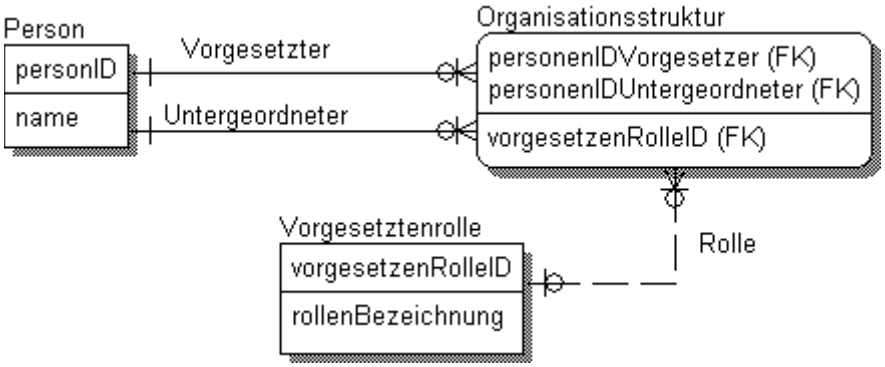


Das obere Bild zeigt eine hierarchische Organisationsstruktur, in der jeder Mitarbeiter einen Vorgesetzten hat. Das Modell ist sehr flexibel, was die Organisationsstruktur von Firmen abgelangt, weil es keine fest definierte Entitäten für die einzelnen Führungsebenen gibt. Bei einer Änderung der Organisationsstruktur, bei der einem Mitarbeiter ein fachlicher und eine disziplinarischer Vorgesetzer zugeordnet werden soll, ist jedoch eine änderung am Datenmodell notwendig.

Zunächst lässt sich die Änderung durch eine weitere rekursive Beziehung lösen. Das nächste Bild zeigt die Entität Person nach der Änderung. Für jede Vorgesetzenrolle, die eine Person für einen anderen Mitarbeiter einnehmen kann ist eine rekursive Beziehung notwendig.



Dies ist dedoch nur teilweise eine zufriedenstellende Lösung, weil sich dadurch eine Veränderung im Datenmodell ergibt. Es wäre besser, wenn sich die Veränderung nur in den Daten niederschlagen würde. Genau dies leistet das ExplizitAssoziation Pattern.



Für jede fachliche Vorgesetzenrolle existiert ein Datensatz in der Tabelle Vorgesetztenrolle. Kommen weitere Rollen hinzu, z.B. Projektleiterrollen, wird dies einfach durch einen weiteren Satz in der Tabelle Vorgesetzenrolle umgesetzt.

Das Beispiel wurde so gewählt, dass ein Vorgesetzter nur eine Rolle einnehmen kann. D.h. er ist gegenüber einem Mitarbeiter entweder disziplinarischer oder fachlicher Vorgesetzer.

Verwandte Muster

- *Validity Period* (*Zeitliche Gültigkeit*).