

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

Es gelten die gleichen Regelungen zur Abgabe wie für Blatt 1.

1 Service Matching (16 Punkte)

Gegeben sei die Ontologie für den Verkauf von Büchern und Videos aus Abbildung 1 (Erweiterung gegenüber der Ontologie aus der Vorlesung).

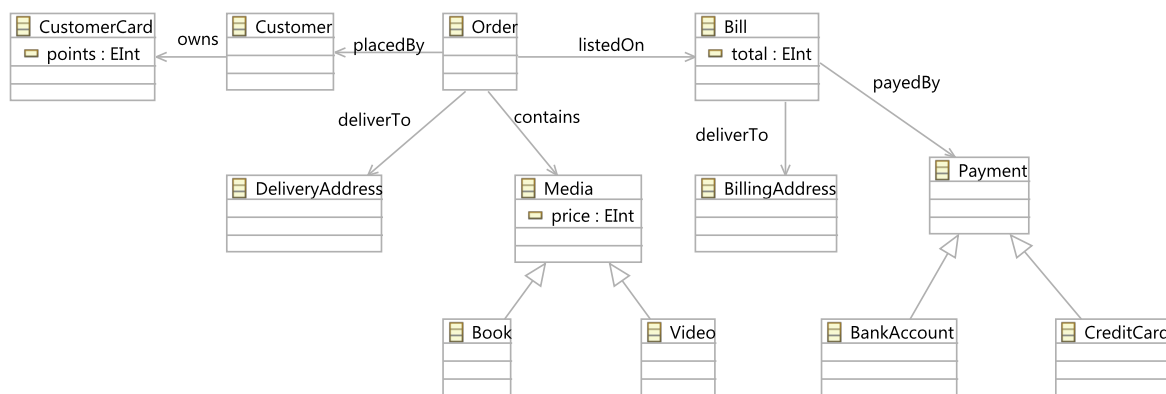


Abbildung 1: Ontologie für den Verkauf von Büchern und Videos

Ein Service Requestor sucht nach den drei Services aus Abbildung 2 (beachten Sie die gegenüber Groove abweichende farbliche Syntax!) und findet einen Provider, der sechs Services anbietet. Die Services sind hierbei wie folgt informell beschrieben:

- **deleteCard_P:** Die Kundenkarte eines bereits im System vorhanden Kunden, eine Kreditkarte und eine Rechnungsadresse werden gelöscht.
- **create_Bill_P:** Es wird eine neue Rechnung erstellt, die eine schon im System vorhandene Rechnungsadresse und Bezahlmethode enthält. Der Rechnungsbetrag ist 0.

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

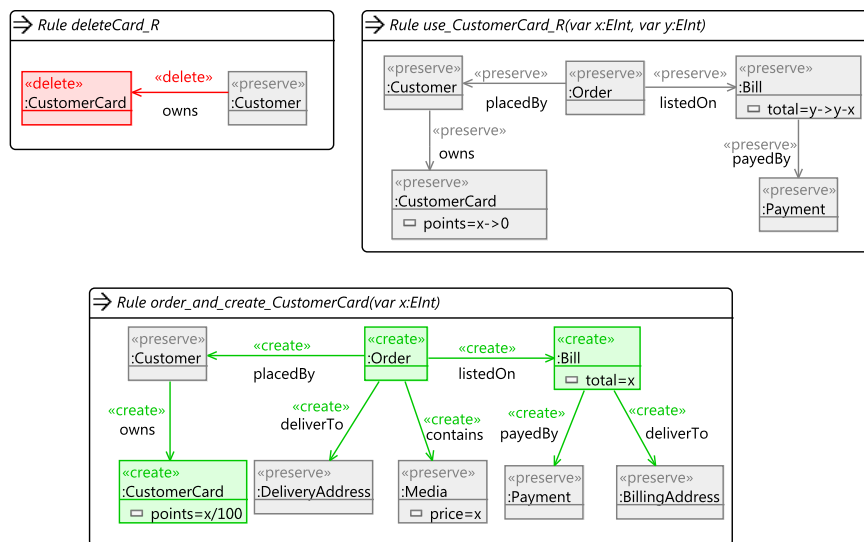


Abbildung 2: Drei Service Requests

- **use_Customer_Card_P**: Es werden keine Elemente erzeugt oder gelöscht. Die Bonuspunkte der Kundenkarte eines Kunden werden vom Betrag einer Rechnung abgezogen. Die Bonuspunkte der Kundenkarte werden auf 0 gesetzt.
- **create_Customer_Card_P**: Ein Kunde erhält eine neue Kundenkarte, die 0 Bonuspunkte enthält.
- **offer_Bonuspoints_P**: Ein Kunde bekommt die Bonuspunkte seiner Bestellung auf seine Kundenkarte gutgeschrieben. Hierbei wird 1% des Rechnungsbetrages zu dem bereits auf der Kundenkarte vorhandenen Betrag hinzugefügt.
- **create_Order_P**: Ein Kunde erstellt eine neue Bestellung. Diese enthält: Den geordneten Artikel, eine Lieferadresse, eine Rechnung, die eine Bezahlmethode und eine Rechnungsadresse enthält. Der Preis des Artikels wird dem bereits auf der Rechnung vorhandenen Betrag hinzugefügt.

Aufgaben:

- Erstellen Sie Regeln für die oben beschriebenen Services und stellen Sie diese grafisch dar. (6 Punkte)

Graph- and Model-Driven Engineering Übungsblatt 3 – Lösungsskizze

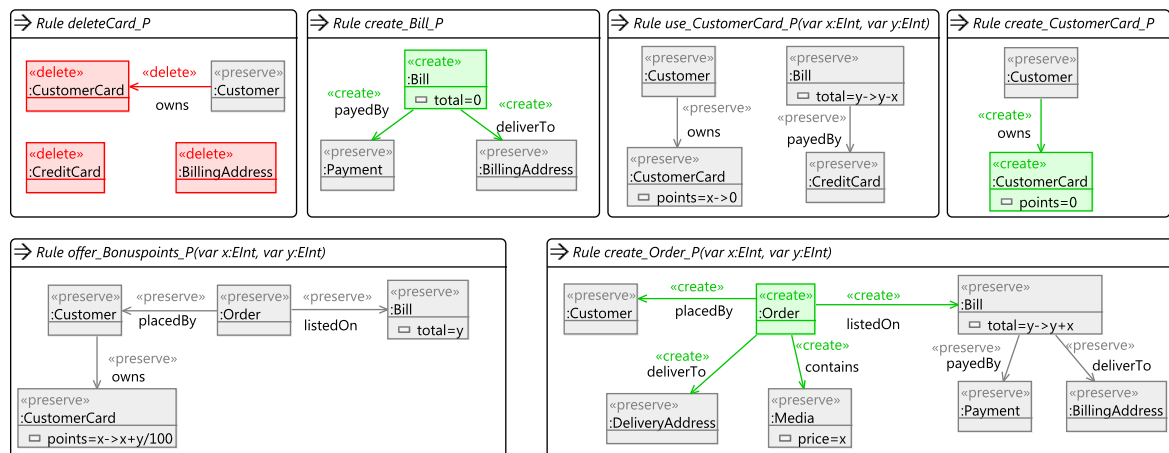


Abbildung 3: Angebote der Provider als Graphtransformationenregeln

- b) Geben Sie für jeden vom Requestor gesuchten Service an, ob dieser *direkt* durch den Provider erfüllt werden kann. Begründen Sie Ihre Antworten informell, anhand der oben beschriebenen Services und geben Sie an, wie die notwendigen Abbildungen zu den von Ihnen in Aufgabenteil a) erstellten Regeln definiert sind bzw. warum die notwendige Abbildung nicht existieren kann. (6 Punkte)
- c) Geben Sie an, ob der Service *order_and_create_CustomerCard* inkrementell durch den Provider mit den von Ihnen erstellten Regeln erfüllt werden kann. Falls ja, begründen Sie Ihre Antwort, indem Sie die beteiligten Schritte nennen und angeben, wie die jeweils notwendigen Abbildungen definiert sind. Argumentieren Sie andernfalls, warum dies mit Ihren Regeln nicht möglich ist. (4 Punkte)

Lösung

- a) Die Regeln sind in Abbildung 3 abgebildet.

Achtung: Zum einen weicht das verwendete Farbschema von Groove ab (hier: rot = Löschung). Zum anderen ist die Lösung so konzipiert, dass damit bestimmte Phänomene nochmal gezeigt werden können; es tauchen Elemente in den Regeln

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

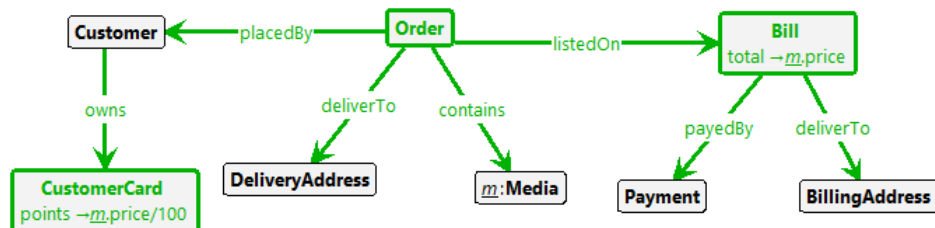


Abbildung 4: Ausgangsregel *order_and_create_CustomerCard*

auf, die man nach der Beschreibung nicht unbedingt bräuchte (z. B. die Verwendung des konkreten Typen **CreditCard** in *use_CustomerCard*).

- b)
- Der Service-Request *deleteCard_R* kann nicht erfüllt werden, da für den einzigen Service, der in der Lage wäre, einen Knoten vom Typ *CustomerCard* zu löschen, *deleteCard_P*, keine Abbildung von der linken Regelseite des Services zum Service Request besteht. Die Knoten **CreditCard** und **BillingAddress** können nicht abgebildet werden (und es gibt auch keine Services, die diese Knoten schaffen, sodass ein inkrementelles Erfüllen möglich wäre).
 - Der Service-Request *use_CustomerCard_R* kann nicht erfüllt werden, da für den einzigen Service, der in der Lage wäre, die Attribute **points** und **total** entsprechend anzupassen, *use_CustomerCard_P*, keine Abbildung von der linken Regelseite des Services zum Service Request besteht. Der Knoten vom Typ **CreditCard** darf nicht auf seine Oberklasse **Payment** gematcht werden.
 - Der Service-Request *order_and_create_CustomerCard* kann nicht direkt erfüllt werden. Offensichtlich gibt es keinen Service, der sowohl eine **Order**, eine **CustomerCard** und eine **Bill** erstellt.
- c) Inkrementell kann der Service-Request *order_and_create_CustomerCard* erfüllt werden. Nacheinander matcht man die Regeln *create_CustomerCard_P*, *create_Bill_P*, *create_Order_P* und *offer_Bonuspoints_P* (partiell). Die einzig möglichen Matches (durch Typisierung in jedem Fall eindeutig festgelegt) entsprechen jeweils der Anforderung an einen (partiellen) Service Match und am Ende sind die zu erzeugenden Elemente vollständig überdeckt. Die Abbildungen 4 bis 8 zeigen den Ablauf.

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

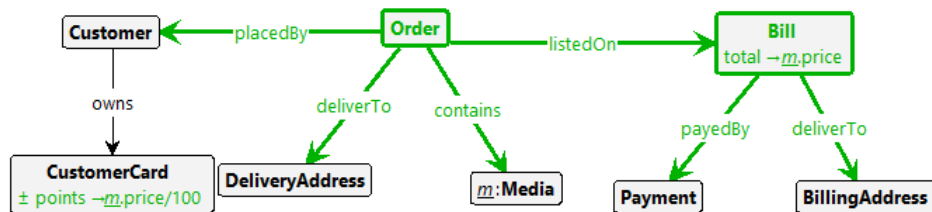


Abbildung 5: Zustand nach partiellem Service Match mit *create_CustomerCard_P*

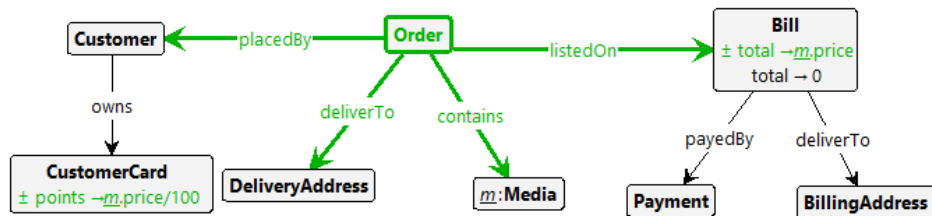


Abbildung 6: Zustand nach partiellem Service Match mit *create_Bill_P*

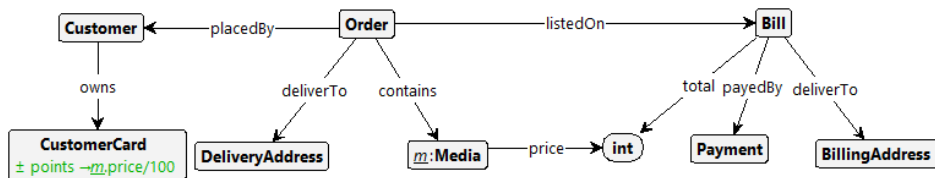


Abbildung 7: Zustand nach partiellem Service Match mit *create_Order_P*

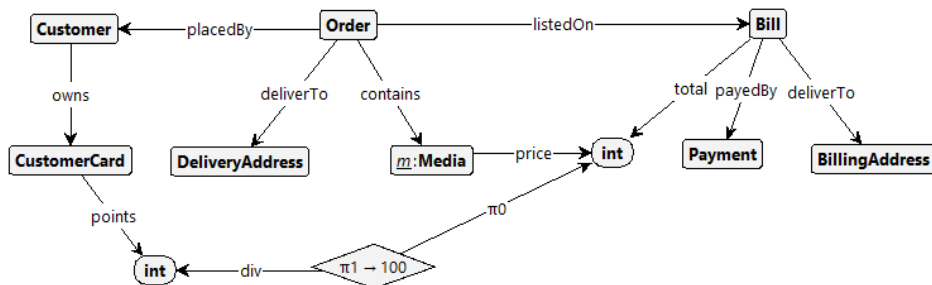


Abbildung 8: Zustand nach Service Match mit *offer_Bonuspoints_P*

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

$$\exists \left(\boxed{o:\text{Order}}, \neg \exists \boxed{o:\text{Order}} \text{--contains--} \boxed{v:\text{Video}} \right)$$

Abbildung 9: Geschachtelte Graphformel in vereinfachter Darstellung

2 Darstellung und Verständnis von Graphformeln (6 Punkte)

Wir betrachten weiterhin die Ontologie (den Typgraphen) für den Verkauf von Medien aus Aufgabe 1 (Abbildung 1).

Aufgaben:

- a) Gegeben ist die geschachtelte Graphformel in vereinfachter Darstellung aus Abbildung 9.
 - i) Erklären Sie in natürlicher Sprache, was die Formel bedeutet. (1 Punkt)
 - ii) Geben Sie die Graphformel in vollständiger Darstellung an. (2 Punkte)
- b) Gegeben ist die geschachtelte Graphformel in vollständiger Darstellung aus Abbildung 10.
 - i) Geben Sie die Graphformel in vereinfachter Darstellung an. Achten Sie darauf, alle Vereinfachungen, die möglich sind, auch anzuwenden. (2 Punkte)
 - ii) Erklären Sie in natürlicher Sprache, was die Formel bedeutet. (1 Punkt)

Lösung

- a) Die Formel besagt, dass es eine Bestellung gibt, die kein Video enthält. Die Formel in vollständiger Darstellung befindet sich in Abbildung 11.
- b) Die Formel in vereinfachter Darstellung befindet sich in Abbildung 12. Die Formel besagt, dass es für jede Bestellung, die auf einer Rechnung gelistet ist, die über einen BankAccount bezahlt wurde, einen Kunden gibt, der die Bestellung durchgeführt hat und der eine Kundenkarte besitzt. (Mit naheliegenden Annahmen an die Multiplizitäten, z. B. nur ein Kunde pro Bestellung, besagt die Formel, dass nur Kunden mit Kundenkarte per Überweisung zahlen können.)

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

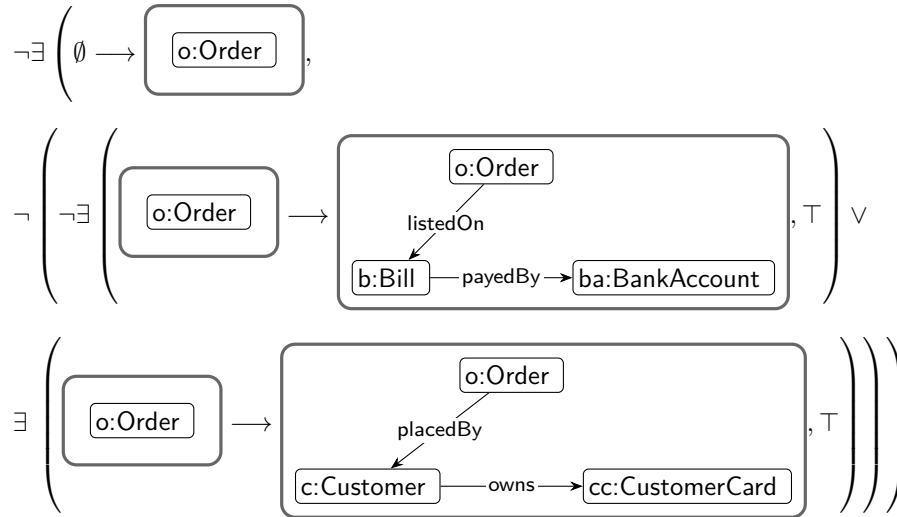


Abbildung 10: Geschachtelte Graphformel in vollständiger Darstellung

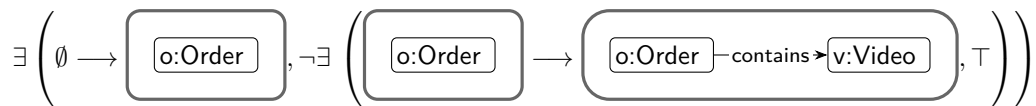


Abbildung 11: Geschachtelte Graphformel aus Abbildung 9 in vollständiger Darstellung

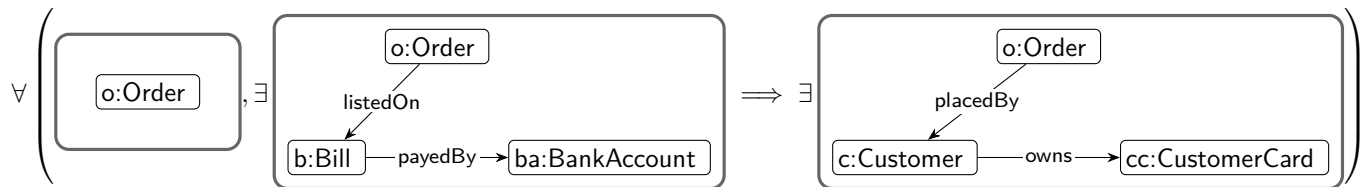


Abbildung 12: Geschachtelte Graphformel aus Abbildung 10 in vereinfachter Darstellung

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

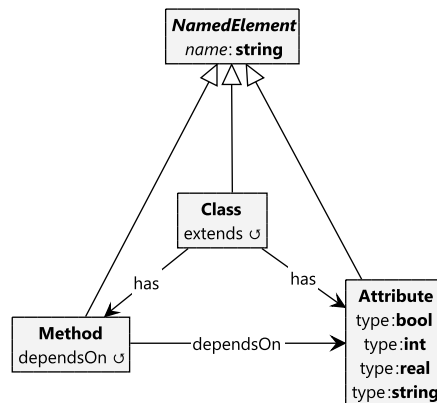


Abbildung 13: Typgraph für vereinfachte Klassendiagramme

3 Entwurf und Auswertung von Graphformeln (8 Punkte)

Gegeben sei der Typgraph für Klassendiagramme von Aufgabenblatt 2 (Sie dürfen Ihre eigene Lösung verwenden oder die in Abbildung 13 gezeigte).

Aufgaben:

- a) Formalisieren Sie die folgenden Aussagen als (über dem gewählten Typgraphen getypte) geschachtelte Graphformeln. Sie dürfen die Formeln in Groove erstellen oder wie in der Vorlesung eingeführt zeichnen; die besprochenen Vereinfachungen in der Darstellung dürfen Sie dabei benutzen.
 - i) Eine Klasse erbt von höchstens einer anderen Klasse. (2 Punkte)
 - ii) Wenn zwei Methoden sich gegenseitig aufrufen, dann liegen sie in der gleichen Klasse. (2 Punkte)
- b) Geben Sie für die Formel i) von oben jeweils einen (über dem gewählten Typgraphen getypten) Graphen an, der die Formel erfüllt, und einen, der dies nicht tut. Begründen Sie jeweils, warum dies der Fall ist (unter Verweis auf die Semantik Ihrer erstellten Graphformel, nicht der natürlichsprachlichen Bedeutung). (4 Punkte)

Graph- and Model-Driven Engineering
Übungsblatt 3 – Lösungsskizze

Lösung

- i) Die beiden Formeln finden sich in der Groove-Datei `Ueb03_KlassendiagrammeFormeln`.
- ii) Die beiden geforderten Graphen befinden sich in der gleichen Groove-Datei. Es gilt: Eine Formel von der Struktur $c = \neg\exists(\emptyset \rightarrow C_1, \top)$ wird per Definition von einem Graphen G *nicht* erfüllt, wenn G die Formel $d = \exists(\emptyset \rightarrow C_1, \top)$ erfüllt. Und eine solche Formel erfüllt ein Graph G per Definition, wenn ein injektiver Morphismus $q: C_1 \rightarrow G$ existiert, sodass (i) $q \circ \iota_{C_1} = \iota_G$ (wobei ι_X der Morphismus vom leeren Graph nach X ist) und (ii) $q \models \top$. Da per Definition jeder Morphismus \top erfüllt und es exakt einen Morphismus vom leeren Graphen nach G gibt, erfüllt ein Graph G eine Formel wie d genau dann, wenn ein injektiver Morphismus von C_1 nach G existiert. Also erfüllt G die Ausgangsformel c genau dann, wenn *kein* injektiver Morphismus von C_1 nach G existiert. Dass dies in den Beispielgraphen einmal der Fall ist und einmal nicht, ist leicht zu sehen.