

Modelltransformatoren - Einführung und Einordnung

Udo Kelter

30.06.2011

Zusammenfassung dieses Lehrmoduls

Bei der modellbasierten Entwicklung müssen bei vielen Gelegenheiten Modelle transformiert werden. Es liegt nahe, hierzu allgemeine Modelltransformationswerkzeuge, die i.w. Graphtransformatoren sind, zu nutzen. Dieses Lehrmodul analysiert die wichtigsten Stellen, an denen in Entwicklungsumgebungen Modelltransformatoren eingesetzt werden können. Ferner wird eine erste Einführung in die grundlegenden Funktionsprinzipien von Modell- bzw. Graphtransformatoren gegeben.

Vorausgesetzte Lehrmodule:

obligatorisch: – Modellgetriebene Software-Entwicklung
empfohlen: – Modellierung graphartiger Dokumente

Stoffumfang in Vorlesungsdoppelstunden: 0.5

Inhaltsverzeichnis

1	Arten von Modelltransformationen	3
1.1	Konversionen zwischen Repräsentationen	3
1.1.1	Externe Ausgabemedien	3
1.1.2	Persistente Speichermedien	4
1.1.3	Konversion von interner zu externer bzw. Persistenter Repräsentation	5
1.1.4	Konversion von externer zu interner Repräsentation	5
1.1.5	Konversion von persistenter zu interner Repräsentation	5
1.2	Übersetzungen auf ein “niedrigeres” Sprachniveau	6
1.3	Editieroperationen	6
2	Exkurs: Semi-Thue-Systeme und Chomsky-Grammatiken	6
2.1	Semi-Thue-Systeme	6
2.2	Chomsky-Grammatiken	8
3	Graphersetzungssysteme und Graphgrammatiken	9
3.1	Grundidee	9
3.2	Varianten der Graphersetzungssysteme	9
3.3	Anwendung	10
	Literatur	10
	Index	10

1 Arten von Modelltransformationen

1. Konversionen zwischen Repräsentationen eines Modells:
 - interne “abstrakte” Laufzeitdarstellung
 - Darstellung auf externen (interaktiven) Ausgabemedien
 - Darstellung in persistenten Speichern
2. Übersetzungen auf ein “niedrigeres” Sprachniveau, Implementation von plattformspezifischen Details
3. komplexe Editieroperationen (Refactorings)

1.1 Konversionen zwischen Repräsentationen

1.1.1 Externe Ausgabemedien

Arten von Repräsentationen von Modellen auf Ausgabemedien:

- textuell: Ausgabemedium = Sequenz von Zeilen;
Alphabet = Zeichensatz
- Hypertext (HTML): Ausgabemedium = Hypertext,
Alphabet = Zeichensatz, Links,
- graphisch: Ausgabemedium = Zeichenfläche;
Alphabet = graphische Symbole und Zeichensatz

–

Manche externen Repräsentationen sind **interaktiv**: Ausblenden von Teilen, Anzeigezustand ggf. pro Modellelement, ...

Große Teile der Ausgabe sind **Layout**: Leerraum, Details graphischer Symbole, Stützpunkte von Linien, ...

- ist semantisch irrelevant
- kann generiert oder manuell erstellt werden
- abhängig vom Ausgabemedium
 - muß vom eigentlichen Inhalt der Modelle getrennt werden, wenn verschiedene Ausgabemedien unterstützt werden

Konversion von interner zu externer Repräsentation: Varianten:

- direkter Erzeugung einzelner Ausgabeelemente über ein API
- mittels einer internen Repräsentation der Graphik (“Graphik-Objekte”):
 1. Erzeugen
 2. durch ein einziges Kommando interpretierenggf. erhebliche Strukturdifferenzen beider Darstellungen

1.1.2 Persistente Speichermedien

i.w. analog zu externen Ausgabemedien

Textuelle persistente bzw. externe Darstellungen sind oft identisch, d.h. Transformatoren sind für beide Zwecke brauchbar

1.1.3 Konversion von interner zu externer bzw. Persistenter Repräsentation

Layout generiert oder manuell?

- generiert: Konversion erzeugt vor allem Layoutdaten
- manuell: Konversion in zwei Schritten:
 1. Integration der Modell- und Layoutdaten;
 2. eigentliche Erzeugung der externen Repräsentation

1.1.4 Konversion von externer zu interner Repräsentation

notwendig in Editoren

Änderungen können zunächst wirksam werden:

1. in den Graphik-Objekten
2. in der Laufzeitrepr. des Modells

Fall 1: erfordert “Interpretation” / Rückwärtstransformation der Veränderung in den Graphik-Objekten

- bei textuellen Darstellungen: z.B. inkrementelle syntaxgesteuerte Editoren bzw. Übersetzer
- bei graphartigen Darstellungen: einfach, wenn konzeptuelle Objekte durch genau ein graphisches Objekt repräsentiert werden;
aber: diverse lokale textuelle Darstellungen (Parameterlisten, Beschriftungen von Transitionen, Multiplizitäten, ...)

Fall 2: einfacher, weil nur erneut unparst werden muß

Hauptnachteil: umständlich zu bedienen, wenn Struktur des abstrakten Syntaxgraphen und Struktur der Graphik nicht identisch

1.1.5 Konversion von persistenter zu interner Repräsentation

i.w. “vordere Hälfte” eines Compilers

- bei XML-Formaten: Konversion in Eingabebaum (-Graph) durch Standardsoftware, danach eigentliche Konversion durch Graphtransformator möglich
- bei frei definierter Syntax: auch Scanner + Parser

1.2 Übersetzungen auf ein “niedrigeres” Sprachniveau

Beispiel: EMF

Typ-orientiert, d.h. alle Instanzen bestimmter Modellelementtypen werden gleichartig transformiert

Können als Graphtransformationen aufgefaßt werden

1.3 Editieroperationen

motiviert vor allem durch Refactorings, die nichtlokale Wirkungen haben

Werden nur an einer Stelle gezielt angewandt
haben i.d.R. Eingabeparameter, die

- ein konkrete Stelle im Modell selektieren
- die ggf. neue Werte festlegen

Können als Graphtransformationen aufgefaßt werden

2 Exkurs: Semi-Thue-Systeme und Chomsky-Grammatiken

2.1 Semi-Thue-Systeme

Semi-Thue-Systeme: sind *Wortersetzungssysteme* (“Texttransformatoren”) und Basis für Chomsky-Grammatiken

Ein **Semi-Thue System (STS)** ist definiert durch:

- ein Alphabet A
- Menge S von Substitutionsregeln der Form $l \rightarrow r$ mit $l, r \in A^*$

Anwendbarkeit einer Regel in einem Wort. Sei
 $w \in A^*$ ein Wort über dem Alphabet A ,
 $s = l \rightarrow r$ eine Substitutionsregel, $s \in S$.
 s ist *anwendbar* auf w , wenn $w = alb$ für passende $a, b \in A^{*1}$

Beispiel: sei

$A = \{ '1', '2', '3', '4' \}$,

$w = '123223'$ und

$s = '23' \rightarrow '4'$ die Substitutionsregel

Dann ist s anwendbar auf w , und zwar sogar an 2 Stellen:

- an Position 1, mit $a = '1'$
- an Position 4, mit $a = '1232'$

Die beiden Positionen bezeichnen wir als **Vorkommen der linken Seite** der Regel s in w .

Übungsaufgabe.

Schreiben Sie ein Java-Programm, das für einen gegebenen String w und eine “linke Seite” l die Positionen aller Vorkommen von l in s ausgibt.

Hinweis: benutzen Sie eine geeignete Variante der vordefinierten String-Funktion `indexOf`.

Anwendung einer Regel an einer Position. Sei

$s \in S$, $s = l \rightarrow r$.

$w = alb \in A^*$, d.h. l kommt in w an Position $|a|$ vor

Die **Anwendung von s auf w an Position $|a|$** liefert das Ergebnis $arb \in A^*$

Übungsaufgabe:

Schreiben Sie ein Java-Programm, das für einen gegebenen String w , eine “linke” und eine “rechte Seite” l und r und eine Position p einen String ausgibt, der aus w durch Anwendung von $l \rightarrow r$ an der Position p entsteht.

¹Hintereinanderschreibung der Unbekannten = Konkatenation

einschrittige Ableitungsrelation: Sei

S die Regelmeng

$w1, w2 \in A^*$.

$w2$ ist bei S **direkt ableitbar** aus $w1$, notiert als $w1 \Longrightarrow_S w2$, wenn es

- eine Regel $s \in S$ und eine Position p in $w1$, an der die Regel s anwendbar ist, gibt und
- wenn die Anwendung von s an der Position p $w2$ ergibt.

allgemeine Ableitungsrelation: Sei

S die Regelmeng

$w1, w2 \in A^*$.

$w2$ ist bei S **ableitbar** aus $w1$, notiert als $w1 \Longrightarrow_S^* w2$, wenn $w1 = w2$

oder wenn $w1 \Longrightarrow_S w2$

oder wenn $v1, \dots, vn \in A^*$ existieren mit $w1 \Longrightarrow_S v1 \Longrightarrow_S \dots \Longrightarrow_S vn \Longrightarrow_S w2$

Anders gesagt ist \Longrightarrow_S^* die reflexiv-transitive Hülle von \Longrightarrow_S .

2.2 Chomsky-Grammatiken

Chomsky-Grammatiken beschreiben Mengen von Texten (Folgen von Zeichen) über einem Alphabet

Eine Chomsky-Grammatik G besteht aus

- einem Semi-Thue-System (A, S) , bei dem das Alphabet in zwei disjunkte Teilmengen N und T aufgeteilt ist, die als nichtterminale bzw. terminale Symbole bezeichnet werden
- einem sog. Startsymbol $s \in N$

Die zur G gehörige Mengen von Texten (Sprache) L ist:

$$L = \{ w \mid w \in T^* \text{ und } s \Longrightarrow_S^* w \}$$

3 Graphersetzungssysteme und Graphgrammatiken

3.1 Grundidee

Graphersetzungssysteme sind direkte Verallgemeinerungen von Wortsatzungssystemen bzw. Chomsky-Grammatiken auf Graphen

- Grundmenge (statt Worten über einem Alphabet): Graphen
- Substitutionsregeln:
 - linke und rechte Seite jeweils ein Graph,
 - Korrespondenzen zwischen Knoten bzw. Kanten der beiden Seiten, die unverändert bleiben und einen Kontext bilden (Korrespondenzen können direkt oder indirekt ausgedrückt sein)
- Vorkommen der linken Seite L (“Mustergraph”) in einem Arbeitsgraphen G: L kommt als Teilgraph in G vor (Teilgraphen-Isomorphie)
- Anwenden einer Substitutionsregel bei einem Vorkommen der linken Seite:
 1. Löschen der Knoten und Kanten der linken Seite, die nicht zum Kontext gehören (also keine Korrespondenz zur rechten Seite der Regel haben);
 2. Kopieren der Knoten und Kanten der rechten Seite in diesen Kontext.

Graphgrammatik: Graphersetzungssystem + Startgraph (analog zum Startsymbol)

3.2 Varianten der Graphersetzungssysteme

- verschiedene Varianten / Typen von Graphen:
 - gerichtet / ungerichtet,
 - einfacher / Multigraph,
 - geordnet / ungeordnet / gemischt,
 - ...

meistens: gerichtet, attributiert und getypt

Typen festgelegt in Graphschema (\sim Metamodell)

- ggf. zusätzliche Unterscheidung von terminalen und nichtterminalen Bestandteilen in den Substitutionsregeln
- Struktur der Regeln (SPO/DPO Ansatz), Nichtanwendbarkeitskonstrukte
- Kontrolle der Regelanwendung: welche Regel, welches Vorkommen?
 - Prioritäten,
 - Kontrollstrukturen

Detailprobleme:

- können “hängende” Kanten entstehen?
- können Regeln parallel ausgeführt werden?

3.3 Anwendung

1. Spezifikation von Editierschritten
2. Spezifikation von Graph- bzw. Modelltransformatoren
3. mit Graphgrammatiken: Spezifikation von korrekten Modellen (aus dem Startgraph ableitbare Graphen)
Problem: effiziente Erkennung
4. diverse Spezialzwecke

Systeme: AGG, Henshin, ATL, GrGen, Progres, VIATRA, XSLT??, QVT (Query/View/Transformation)

weitere s.

http://en.wikipedia.org/wiki/Model_Transformation_Language

Literatur

- [1] Model Transformation Language; 2011; http://en.wikipedia.org/wiki/Model_Transformation_Language