

# A Graphical Specification of Model Transformations with Triple Graph Grammars

Lars Grunske<sup>1</sup>, Leif Geiger<sup>2</sup>, and Michael Lawley<sup>3</sup>

<sup>1</sup> School of Information Technology and Electrical Engineering,  
University of Queensland, Brisbane, QLD 4072, Room 72-458 IT Building  
grunske@itee.uq.edu.au

<http://www.itee.uq.edu.au/>

<sup>2</sup> University of Kassel, Software Engineering Research Group,  
Department of Computer Science and Electrical Engineering,  
Wilhelmshöher Allee 73, 34121 Kassel, Germany  
leif.geiger@uni-kassel.de

<http://www.se.eecs.uni-kassel.de/se/>

<sup>3</sup> CRC for Enterprise Distributed Systems Technology (DSTC)\*\*,  
University of Queensland,  
Brisbane, QLD 4072, Australia  
michael@lawley.id.au  
<http://www.dstc.edu.au/>

**Abstract.** Models and model transformations are the core concepts of OMG's MDA<sup>TM</sup> approach. Within this approach, most models are derived from the MOF and have a graph-based nature. In contrast, most of the current model transformations are specified textually. To enable a graphical specification of model transformation rules, this paper proposes to use triple graph grammars as declarative specification formalism. These triple graph grammars can be specified within the FUJABA tool and we argue that these rules can be more easily specified and they become more understandable and maintainable. To show the practicability of our approach, we present how to generate Tefkat rules from triple graph grammar rules, which helps to integrate triple graph grammars with a state of the art model transformation tool and shows the expressiveness of the concept.

## 1 Introduction

Model Driven Engineering (MDE) is a software engineering principle that promotes the use of models and transformations as primary development artifacts. To practically apply this principle, the Object Management Group (OMG) has proposed the MDA<sup>TM</sup> [1] as a set of standards for integrating MDE tools. The MDA<sup>TM</sup> approach separates the specification of systems from the implementation of these systems. For this reason two basic model types are introduced,

\*\* The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme.

## Titel und Autor

Euer Paper sollte mit einem ausdrucksstarken Titel beginnen, der knapp den Inhalt des Papers beschreibt. Seid spezifisch und benutzt Schlüsselwörter mit einer starken Assoziation zum Inhalt, da die meisten Leser euer Paper über Web- oder Literaturliteraturdatenbank-Recherche finden werden, wo meist der Titel indexiert ist.

### Aufbau

Unter dem Titel folgen (typischerweise mit doppeltem Zeilenabstand) die Autoren, darunter die Instituts-/Organisationszugehörigkeit. Das kann wie hier die volle Adresse mit Mail und Webseite sein, oder häufig auch nur sowas wie „University of Kassel“

## Abstract

Der Abstract ist eine Kurzzusammenfassung (1-2 Absätze, 150-250 Wörter) eures Papers. Er sollte alle Kernaspekte des Papers kurz, knapp und vor allem verständlich wiedergeben und Lesern dabei helfen, zu entscheiden, ob das Paper für sie interessant ist (das ist „im echten Leben“ insbesondere bei Buch- oder Journal-Beiträgen wichtig, für die häufig nur der Abstract frei verfügbar ist).

### Aufbau

Der Abstract folgt typischerweise einer klaren Struktur:

- *Problembeschreibung:*

- optional: Kontext, Problemdomäne - je nach Scope der Konferenz und der zu erwartenden Leserschaft sollte der Kontext des Papiers kurz beschrieben werden

- Problem, Zweck des Papers, Fragestellung - Was ist der Haupt-Knackpunkt, um den es geht? Warum ist das interessant/wichtig?

- *Lösungsansatz:*

- kurze, prinzipielle Erklärung des Lösungswegs

- *Ergebnis:*

- Antwort auf die Problem-Fragestellung

- Interpretation des Ergebnisses

Kent Beck beschreibt in [1] den Abstract etwas anders und lockerer, im Endeffekt läuft es aber auf dasselbe hinaus.

Für eure Seminaarausarbeitung, die nicht so sehr eigene Forschungsergebnisse, sondern existierende Techniken beschreibt, sollte der Lösungsansatz prinzipiellen Zweck und Funktion der untersuchten Technik umfassen. Das Problem sollte zeigen, warum man die Technik braucht und das Ergebnis, wie der Einsatz der Technik hilft.

[1] Kent Beck, How to Get a Paper Accepted at OOPSLA, <http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html>

Kontext

Problem

Lösungsansatz

Interpretation  
(Hier nicht so schön)

Ergebnis