

# Zusammenfassung

Modelltransformationen spielen eine Schlüsselrolle in automatisierten Software-Entwicklungsprozessen. Der moderne Trend geht in Richtung Spezifikation der Software mit einem abstrakten Modell und dessen schrittweiser Verwandlung zum Programmiercode. Es ist sehr wichtig, dass der Programmiercode der ursprünglichen Spezifikation entspricht. Deswegen ist eine wichtige Frage einer solchen Modelltransformation, ob das transformierte Modell die Verhaltenseigenschaften des Ausgangsmodells erfüllt.

In dieser Doktorarbeit wird eine neue Methode für die Spezifikation von einer Modelltransformation entwickelt und es wird gezeigt, dass die zu Grunde gelegte Modelltransformation korrekt ist. Dabei bedeutet Korrektheit, dass die Verhaltenseigenschaften bei der Modelltransformation erhalten bleiben. Die Methode wird in dem Ansatz der modellgetriebenen Architektur eingeführt, wo ein Metamodell für modellierende Sprachen definiert wird. Ein Metamodell ist spezifiziert anhand objektorientierter Ansätze, wo die Diagramme auf Graphen basieren und sind manchmal mit textuellen Informationen angereichert. Von daher kann ein Metamodell auch als ein Graph bezeichnet werden. Die Modelltransformation ist dann anhand von Triple Graph Grammatik definiert.

In dieser Doktorarbeit werden Sprachen benutzt, die Verhaltenssemantik beinhalten, welche wiederum mit der Graph Transformation System Methode definiert sein kann. Hierin wird die Graph Transformation zweimal verwendet: Spezifikationen für Modelltransformation und für Verhaltenssemantik. Als Ergebnis der Anwendung der Graph Transformationen zu dem Graph, der ein Modell bezeichnet, wird ein Labelled Transition System (LTS) erzeugt. Ein LTS vertritt einen Satz von allen möglichen Transitionsfolgen, die das Systemverhalten beschreiben sollen.

Eine temporale Logik, die ACTL (Aktionbasierte Logik von CTL\*) genannt wird, wird verwendet, um die Verhaltenseigenschaften des Modells in ein LTS zu spezifizieren. Vollständige Erhaltung der Verhaltenseigenschaften bedeutet, dass ein LTS, welches für *jede* Instanz von Modell erzeugt wurde, und weiteres LTS, generiert für ein korrespondierendes Zielmodell, können nicht unterschieden werden durch den Wahrheitswert von ACTL Formeln. Wenn beide LTS die gleichen ACTL Eigenschaften aufweisen, dann werden diese LTSs ACTL äquivalent genannt.

Die Methode basiert auf dem Ansatz einer Äquivalenzbeziehung (schwache Bisimulation) über LTS, welche das Standardkonzept in der concurrency Theorie ist. Normalerweise benutzt man diese Äquivalenzbeziehung um Berechnungen zu vergleichen und zu entscheiden, ob sie tatsächlich äquivalent sind oder nicht. In dieser Doktorarbeit wird gezeigt, dass schwache Bisimulation auf LTS die ACTL Äquivalenz impliziert. Um die Erhaltung der Verhaltenseigenschaften während der Modelltransformationen zu reifen, ist es erforderlich der schwache Bisimulation auf

LTS von Anfangs- und Ausgangsmodellen zu erstellen.

Zusammenfassend bietet die Methode eine Leitlinie, wie Modellierungssprachen und Modelltransformationen spezifiziert werden können, damit es möglich ist zu zeigen, dass spezifizierte Modelltransformation semantisch korrekt ist. Sie führt einen Beweis aus der zeigt, dass die definierte Äquivalenz eine schwache Bisimulation ist. Zusätzlich wird erklärt wie die Verhaltenseigenschaften eines Ausgangsmodell für eine Zielsprache interpretiert werden können durch die Benutzung von Triple Graph Grammatik Transformation und der schwache Bisimulation.

Ferner wurde die Methode mit einer Fallstudie bestätigt, wo eine Modelltransformation zwischen zwei Modellsprachen definiert wurde: Calculus of Communication Systems (CCS) und Petri Netze. Das Hauptziel der Fallstudie ist, zu zeigen, wie man beweist, dass die definierte Modelltransformationen das Semantikverhalten beibehält. Sie illustriert die Spezifikation der Verhaltenseigenschaften für CCS und die Interpretation bezüglich der Petri-Netz-Sprache.