

# Generierung konsistenzerhaltender Editierskripte im Kontext der Modellversionierung

Timo Kehrer  
Praktische Informatik, Universität Siegen  
kehrer@informatik.uni-siegen.de

**Abstract:** Modellbasierte Softwareentwicklung erfordert spezialisierte Werkzeuge für ein professionelles Versions- und Variantenmanagement von Modellen. Insbesondere Anwendungsfälle wie das Patchen oder Mischen von Modellen stellen sehr hohe Anforderungen an die Konsistenz der synthetisierten Modelle. Ein Lösungsansatz ist die Verwendung konsistenzerhaltender Editierskripte. Zentrale Herausforderung ist letzten Endes die Generierung solcher Editierskripte, welche wir in diesem Papier mit entsprechenden Hinweisen auf weiterführende Literatur kurz skizzieren.

## 1 Motivation

Modellbasierte Softwareentwicklung hat sich in einigen Applikationsdomänen inzwischen fest etabliert. Modelle sind hier primäre Artefakte, entwickeln sich daher ständig weiter und existieren im Laufe ihrer Evolution in zahlreichen Versionen und Varianten. In der Praxis zeigt sich sehr deutlich, dass man für Modelle die gleichen Versionsmanagement-Dienste benötigt, die man für textuelle Dokumente gewohnt ist, namentlich Werkzeugfunktionen zum Vergleichen, Patchen und Mischen von Modellen.

Derzeitig verfügbare Werkzeuge des Versions- und Variantenmanagements von Modellen arbeiten jedoch auf systemnahen, fallweise werkzeugspezifischen Repräsentationen von Modellen und unterstellen ferner generische Graphoperationen zur Beschreibung von Änderungen. Dies führt zu zwei wesentlichen Problemen:

1. Die Darstellung solcher “low-level” Änderungen ist meist unverständlich, ohne Kenntnisse der internen Repräsentation der Modelle teilweise sogar unmöglich.
2. Die Anwendung von low-level Änderungen in Patch- oder Mischszenarien birgt die Gefahr der Synthetisierung inkonsistenter Modelle, da i.d.R. nicht alle Änderungen auf das Zielmodell propagiert werden können. Im schlimmsten Fall kann ein Modell so inkorrekt werden, dass es nicht mehr mit Standard-Modelleditoren verarbeitet werden kann.

## 2 Modellversionierung auf Basis von Editieroperationen

Lösungsansätze für die vorstehend skizzierten Probleme werden im Forschungsprojekt MOCA<sup>1</sup> erarbeitet. Ziel des Projekts ist es, alle für das Versions- und Variantenmanagement relevanten Werkzeugfunktionen auf die Abstraktionsebene von Editieroperationen anzuheben, welche für einen Benutzer *verständlich* und bei der Anwendung auf ein Modell *konsistenzerhaltend* sind.

**Konsistenzerhaltende Editieroperationen.** Unter einer konsistenzerhaltenden Editieroperation verstehen wir eine in-place Transformationsvorschrift, welche ein konsistentes Modell in einen konsistenten Folgezustand überführt, unabhängig von den aktuellen Aufrufparametern der Operation. Ein Modell bezeichnen wir als konsistent, wenn es konform zum effektiven Metamodell der unterstellten Editierumgebung ist. Die Menge der zulässigen Editieroperationen hängt somit vom Modelltyp und letzten Endes auch von der gegebenen Editierumgebung ab.

**Editierskripte.** Bei der Propagation von Änderungen in Patch- oder Mischszenarien sollen Änderungsvorschriften benutzt werden, welche ausschließlich aus Aufrufen von Editieroperationen bestehen. Solch eine halbgeordnete Menge von Operationsaufrufen bezeichnen wir als *Editierskript* (engl.: *Edit Script*). Zentrale Grundlage für die Realisierung entsprechender Patch- und Mischwerkzeuge ist also die Generierung von Editierskripten.

**Generierung von Editierskripten.** Editierskripte sind in der Regel durch Vergleich zweier Versionen eines Modells möglichst effizient zu berechnen. Ausgangspunkt unseres Algorithmus ist eine gegebene low-level Differenz, welche mit existierenden Verfahren des Modellvergleichs erzeugt werden kann. Diese wird nachfolgend in mehreren Schritten zu einem Editierskript weiter verarbeitet: Zunächst werden Gruppen von low-level Änderungen identifiziert, wobei eine Gruppe (engl.: *Semantic Change Set*) den Aufruf einer Editieroperation repräsentiert [KKT11]. Anschließend werden die Argumente der identifizierten Operationsaufrufe extrahiert und die sequentiellen Abhängigkeiten der Operationsaufrufe analysiert [KKT13]. Der Algorithmus arbeitet generisch in dem Sinne, als dass er durch die Eingabe der Spezifikationen der zulässigen Editieroperationen konfiguriert und somit an beliebige Modelltypen und Editorumgebungen angepasst werden kann.

## Literatur

- [KKT11] T. Kehrer, U. Kelter und G. Taentzer. A Rule-Based Approach to the Semantic Lifting of Model Differences in the Context of Model Versioning. In *Proc. 26th IEEE/ACM Intl. Conf. on Automated Software Engineering*. IEEE Computer Society, 2011.
- [KKT13] T. Kehrer, U. Kelter und G. Taentzer. Consistency-Preserving Edit Scripts in Model Versioning. In *Proc. 28th IEEE/ACM Intl. Conf. on Automated Software Engineering*. IEEE, 2013.

---

<sup>1</sup><http://www.dfg-spp1593.de/index.php?id=46>