

Versionisierung für modell-gestützte Softwareentwicklung über Werkzeuggrenzen hinweg

Christian Hein, Tom Ritter
Fraunhofer Institut für offene Kommunikationssysteme
Kaiserin-Augusta-Allee 21, Berlin 10589

Abstract

Dieses Positionspapier stellt eine Zusammenfassung der gegenwärtigen Arbeitsergebnisse die im Fraunhofer Institute FOKUS erzielt wurden dar

1 Einleitung

Versionisierung spielt in der Softwareentwicklung traditionell eine entscheidende Rolle. Basierend auf Quellcode existieren bereits zahlreiche Lösungen wie das Concurrent Versions System (CVS) oder Subversion (SVN).

2 Hintergrund

Die hier vorgestellten Ergebnisse, wurde unter anderem in den Projekten Modelware [1] und ModelPlex [2] erzielt. Modelware hatte ein Laufzeit von 24 Monaten und ein Budget von 20 Millionen Euro. In Modelware wurden in einem gemeinsamen europäischen Aufwand die verschiedenen Forschungsschwerpunkte im Bereich der modell-gestützten Entwicklung adressiert. ModelPlex als Nachfolger des Modelware Projekts mit ebenfalls einem Budget von 20 Millionen Euro vertieft die gewonnen Ergebnisse. Insbesondere die Zusammenarbeit mit Richard Paige, Dimitrios Kolovos (University of York) sowie Nick Dowler und Pete Rivett (Adaptive) ist ausschlaggebend für die hier dargestellten Einschätzungen und Ergebnisse.

3 Problemstellung

Das Konzept der Versionisierung besteht aus zwei Teilbereichen. Zum einen müssen Unterschiede zwischen Modellen erkannt werden und zum anderen muss es möglich sein verschiedene Versionen eines Modells zu einem neuen Modell zusammenzufügen. Für die Erkennung von Differenzen in Modellen gibt es bereits zahlreiche Herangehensweisen und Lösungen. Dem gegenüber ist die Komposition von Modellen ein aktuelles Forschungsthema im Bereich der modell-gestützten Softwareentwicklung. Hierfür existieren verschiedene Ansätze und Begriffe, wie das Verweben von Modellen (Model Weaving) oder das Verschmelzen von Modellen (Model Merging). Ausgehend von diesen Ansätzen betrachten wir

das Zusammenfügen von verschiedenen Modellversionen als Spezialfall des Verschmelzens von Modellen. Jedoch existieren auch Herangehensweisen, die Model Weaving für diesen Teil der Versionisierung von Modellen verwenden. Dessen ungeachtet ist es bei allen Ansätzen schwer eine vollautomatische Komposition von Modellen im Bezug auf Versionisierung zu realisieren. Das Konzept der Versionisierung lässt vermutlich nur halbautomatisierte Ansätze zu.

Darüber hinaus beruht ein modell-gestützter Softwareentwicklungsprozess selten auf einem einzelnen Modell, vielmehr entsteht ein so entwickeltes Softwareprodukt aus einer Vielzahl von unterschiedlichen Modellen, die in unterschiedlichen Phasen der Entwicklung entstehen und auch über Werkzeuggrenzen verändert werden können. Dies führt unabwendbar zu unterschiedlichen Versionen von Modellen im gesamten Entwicklungsprozess. Um verschiedene Versionen eines oder mehrerer Modelle konsistent behandeln zu können ist es wichtig, dass Modellelemente und alle Versionen der Elemente einzeln identifizierbar sind.

4 Globale Herangehensweise

Zur Identifizierung von Modellelementen bietet die OMG mit MOF Versioning [3][4] bereits einen hilfreichen Ansatz. In dieser Spezifikation werden auf M3 Ebene Erweiterungen für MOF definiert, die es ermöglichen Versionisierungsinformationen, wie eindeutige Kennzeichner, an MOF Elemente anzufügen. Dennoch bleibt die Frage wie eine solche Vergabe von eindeutigen Kennzeichnern realisiert werden kann zunächst unbeantwortet. In einfachen Entwicklungsprozessen, in denen die Systementwicklung mittels eines einzelnen Werkzeuges vorgenommen wird, ist diese Frage meist leicht zu beantworten. Nicht selten werden in industriellen Softwareentwicklungsprojekten aber sehr viele verschiedene Werkzeuge von unterschiedlichen Herstellern eingesetzt, die ihrerseits unterschiedlich mit den eindeutigen Bezeichnungen umgehen. Eine mögliche Lösung dieses Problems wird gegenwärtig im Rahmen des ModelPlex Projektes untersucht. Die Idee ist, dass die Integration der verschiedenen Werkzeuge über ein gemeinsames Verständnis, genauer über die Verwendung eines

gemeinsamen Meta-Modelles erreicht werden kann. Die Integration findet dann in Modell-Repositories statt, die dann die Vergabe und Verwaltung von eindeutigen Kennzeichnern übernehmen können.

Unter Voraussetzung, dass eine Vergabe von eindeutigen Kennzeichnern in einem verteilten Entwicklungsprozess erfolgreich war, lassen sich einige Vorteile für den Entwicklungsprozess ableiten. So könnte nicht nur die Versionisierung grundsätzlich gewährleistet werden, sondern auch mit Hilfe der globalen Kennzeichner könnten OCL Ausdrücke über Versionen hinweg formuliert werden. Der Vorteil hierfür wäre, dass bestimmte Eigenschaften nur bestimmten Versionen zugeordnet werden könnten. Ebenso einfach wäre es ein Traceabilitywerkzeug zu realisieren, das Zusammenhänge zwischen verschiedenen Modellen mittels globaler eindeutiger Kennzeichner aufzeichnet.

5 Prototyp eines Versionisierungswerkzeugs

Im Rahmen des Modelware Projektes wurde bereits ein erster Prototyp eines Versionisierungswerkzeugs entwickelt. Abbildung 1 zeigt die Darstellung einer Differenz von zwei verschiedenen Versionen eines UML Modells. Im konkreten Beispielfall wurde der Name eines UML Paketes verändert. Die Arbeitsweise des Werkzeuges sowie der zugrunde liegende Algorithmus zur Auffindung von Unterschieden sind detailliert in [5] beschrieben. Grundsätzlich benutzt das Werkzeug für die Differenzanalyse die Eigenschaft, dass alle Modelle sowie Modellelemente über eindeutige Kennzeichner (MOF ID) identifizierbar sind. Weiterhin benutzt das Werkzeug die reflektierende Eigenschaft der M3 Sprache (MOF), dadurch ist der Algorithmus unabhängig vom konkreten Metamodel und ist beispielsweise nicht beschränkt auf UML Modelle. Diese Unabhängigkeit bietet einige Vorteile, so lassen sich domain-spezifische Modelle ebenfalls versionisieren, die beispielsweise häufig in Systemen des Automobilbereich oder anderen eingebetteten Systemen Verwendung finden.

Als ein weiteres Ergebnis der Entwicklung dieses Werkzeuges ist zu nennen, dass bestimmte Kategorien von Objektveränderungen identifizierbar sind (siehe ebenfalls [5]), die entsprechend visuell dargestellt werden können und sollten.

6 Fazit und Ausblick

Um Versionisierung über Werkzeuggrenzen zu realisieren werden Modell-Repositories benötigt, die die globalen Kennzeichner vergeben. Mindestens jedoch wird eine Interaktion mit einem Versionisierungswerkzeug oder -dienst benötigt. Ebenso denkbar ist die einfachere Berechnung von Metriken über Versionsinformationen, so lässt sich beispielsweise automatisiert ermitteln, wie viele Klasse zu welcher Zeit im Modell vorhanden waren. Darüber

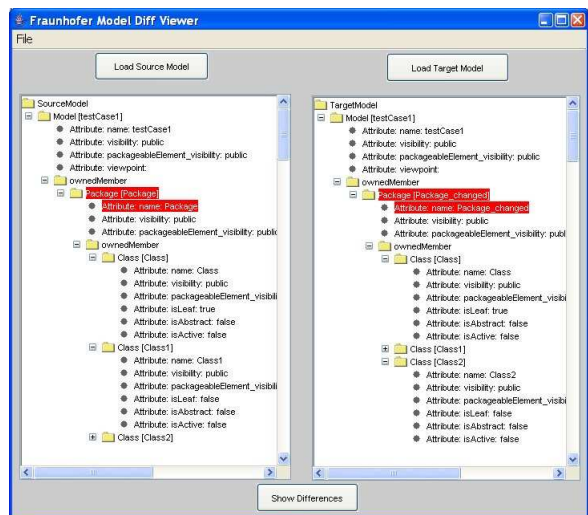


Figure 1: Bildschirmaufnahme vom Fraunhofer Fokus Diff/Merge Tool

hinaus besitzen Modelle in einem komplexen Softwareentwicklungsprozess Beziehungen zu Artefakten. Da die Artefakte sowohl manuell erzeugt als auch aus Modellen abgeleitet werden können, ist es wichtig die Versionisierung auf die zugeordneten Artefakte geeignet zu erweitern.

References

- [1] Modelware (MODELLing solution for softWARE systems), Project co-funded by the European Commission under the IST Sixth Framework Programme (2002-2006), www.modelware-ist.org
- [2] MODELPLEX (MODELLing solution for COMPLEX software systems), Project co-funded by the European Commission under the IST Sixth Framework Programme (2002-2006), www.modelplex.org
- [3] OMG Document, Meta Object Facility (MOF) Core Specification, OMG Available Specification Version 2.0, formal/06-01-01
- [4] OMG Document, Meta Object Facility (MOF) 2.0 Versioning and Development Lifecycle Specification, Final Adopted Specification, ptc/05-08-01
- [5] Engel, Paige, Kolovos, Using a Model Merging Language for Reconciling Model Versions, A. Rensink and J. Warmer (Eds.): ECMDA-FA 2006, LNCS 4066, pp.143-157, Springer 2006