

# Konsistenz und Vollständigkeit industrieller UML Modelle

Christian Lange, Michel Chaudron

Faculteit Wiskunde en Informatica  
Technische Universiteit Eindhoven  
Postbus 513  
5600 MB Eindhoven (Niederlande)  
{C.F.J.Lange, M.R.V.Chaudron}@tue.nl

**Abstract:** Mit steigendem Abstraktionsniveau wächst die Menge der möglichen Interpretationen eines UML Modells. Inkonsistenzen und Unvollständigkeiten verursachen Mehrdeutigkeiten und vergrößern die Menge der möglichen Interpretationen unnötigerweise. Dies gilt es zu verhindern. Dazu haben wir eine Reihe von Regeln entwickelt und in einem Werkzeug implementiert. Industrielle Fallstudien haben gezeigt, dass die auftretende Anzahl der Inkonsistenzen und Unvollständigkeiten in der Praxis erschreckend hoch ist.

## 1 Einsatz von Modellen in der Softwareentwicklung

In der Softwareentwicklung ist der Einsatz von Modellen nicht mehr wegzudenken. Das Erstellen von Modellen trägt zum Problemverständnis bei, Modelle dienen zur Spezifizierung von Softwaresystemen, sie sind Dokumentations- und Kommunikationsmittel und werden zur frühen Analyse von Systemeigenschaften herangezogen. Die UML hat sich durchgesetzt für Designmodelle und wird mehr und mehr in der Architekturmodellierung eingesetzt [SNH99].

Am Beginn des Softwareentwicklungsprozess werden Modelle eines hohen Abstraktionsniveaus eingesetzt; mit zunehmendem Problemverständnis werden die Modelle detaillierter. Dies resultiert letztendlich im Programmcode, der die Lösung – per Definition, er ist kompilierbar – eindeutig beschreibt und nur eine Interpretation zulässt. Mit steigendem Abstraktionsniveau wächst die Menge der möglichen Interpretationen.

## 2 Konsistenz und Vollständigkeit

Die UML besteht aus einer Vielzahl, sich teilweise überschneidender Diagrammtypen. Durch diese Schnittmengen können Inkonsistenzen und Unvollständigkeiten entstehen. Wir unterscheiden Inkonsistenzen mit den „Well-formedness“ Regeln (in [OMG] und erweitert in [La03a]), Inkonsistenzen zwischen Diagrammen und Unvollständigkeiten zwischen Diagrammen.

Inkonsistent zu den „Well-formedness“ Regeln bedeutet eine Abweichung des Modells von der UML Spezifikation bzw. von allgemein anerkannten oder organisationsspezifischen Standards. (Beispiel: eine Klasse ohne Attribute und Operationen)

Inkonsistenzen zwischen Diagrammen treten auf, wenn Modellelemente in verschiedenen Diagrammen widersprüchlich beschrieben sind. (Beispiel: Im Sequenzdiagramm wird eine Operation einer Klasse aufgerufen, die im Klassendiagramm nicht definiert ist).

Unvollständigkeiten zwischen Diagrammen sind zum Beispiel fehlende State Chart Diagramme bei Klassen, die aufgrund ihrer Beschreibung in den Sequenzdiagrammen ein stark dynamisches internes Verhalten haben.

Diese Situationen verursachen Unklarheit und Mehrdeutigkeiten eines Modells und vergrößern somit unnötigerweise die Menge  $I$  aller möglichen Interpretationen. Dies gilt es zu minimieren. In [La03a, La03b] haben wir eine Reihe von Regeln vorgestellt, die dem Auffinden von Inkonsistenzen und Unvollständigkeiten dienen. Diese Regeln sind in einem Werkzeug (SAAT, Software Architecture Analysis Tool) implementiert.

Anhand von Fallstudien (vier industrielle Modelle aus Projekten mit mehr als 100 Mannjahren, zwei Modelle postgraduierter Studenten) haben wir (stichprobenartig) Konsistenz- und Vollständigkeitsprobleme in der Praxis analysiert.

### 3 Fazit

Projekte schreiten mit stark unvollständigen und inkonsistenten Modellen unter Zeitdruck in folgende Phasen. Die Aussagekraft von Qualitätsanalysen unvollständiger Modelle ist fraglich, da sie bei großer Menge  $I$  meist sehr allgemein sind, oder sich auf eine (zufällige) Interpretation beziehen, die nicht unbedingt mit der angestrebten Lösung übereinstimmt.

In zukünftigen Forschungsvorhaben sollte der Zusammenhang zwischen Problemen in Modellen und Problemen in der Implementierung untersucht werden.

### Literaturverzeichnis

- [La03a] Lange, C.: Empirical Investigations in Software Architecture Completeness. Master's Thesis, Technische Universität Eindhoven, September 2003.
- [La03b] Lange, C.; Chaudron, M.R.V.; Muskens, J.; Somers, L.J.; Dortmans, H.M.: An Empirical Investigation in Quantifying Inconsistency and Incompleteness of UML Designs. 2nd workshop on consistency problems in UML-based software development, October 20, 2003, in San Francisco. Part of the "International Conference on UML 2003".
- [OMG] OMG, Hrsg.: OMG Unified Modeling Language Specification, Version 1.5. OMG, 2003. OMG Dokument formal/03-03-01, [www.omg.org](http://www.omg.org).
- [SNH99] Soni, D.; Nord, R.; Hofmeister, C.: Applied Software Architecture. Addison-Wesley Pub Co., 1<sup>st</sup> edition, 1999.