

Software Architekturmodelle

Alexander, Dominik, Niels, Steffen

Semantik von (Meta-)Modellen

- Semantikdefinition?
 - Wie kann man ohne leben????
 - Generierung
 - Kommunikation
 - Analyse
 - Testfallgenerierung
 - Migration/Adaptergenerierung
 - ...

Problem: UML 2

- 1. Möglichkeit:
 - Umfassende Definition für UML2
 - Unrealistisch
- 2. Möglichkeit
 - Jeder definiert stattdessen eigenes Meta-Model
 - Keine Kommunikation
 - Keine allgemeinen Analyse-Tools
 - Evtl. hilft hier Traceability
 - Versionierung der Semantik
 - Auswahl gemeinsamer Standard-Formalismen, in denen die Semantik spezifiziert wird
 - Realisierbar?

Compilerbaulösungsansatz

- Intermediate Language Ansatz?
 - Zwischensprache
 - N Frontendsprachen
 - M Backendsprachen
 - N+M Transformationen
 - Immer nur Tupel von Sprachen müssten bekannt sein
 - Compiler-Warnings / Errors bei Mehrdeutigkeiten
 - Vgl. ANSI C Compiler

Werkzeugdefinierte Prozesse

- Prozessprobleme
 - Kein einheitlicher Prozess
 - Referenzprozess wäre hilfreich

Werkzeuge: Stand der Dinge

- Tools
 - Wozu Semantik wenn Syntax unklar?
 - Inkompatible Tools
 - Konkrete Syntax (Graphik, XMI-Exporte, ...)
 - Abstrakte Syntax (XMI-Exporte, ...)
 - Generatoren

Evaluation von MDA

- Experimente und Studien zu Prozessen und Tools
 - Unternehmen machen es nicht
 - Höchstens Fallstudie
 - Vergleichende Studien wären nötig
 - Soziologische / Psychologische Studien
 - → Wissenschaftlicher Auftrag!
 - ABER: Viele Probleme erst bei realistischen (groß, undokumentiert, alt, unklar, business-critical) auftretend, nahezu nicht nachbildbar in univer. Experiment
 - Experimentelle Methodik?

Vorwissen / Ausbildung

- Ausbildung?
 - Software Architekturmodelle und Semantiken(!) unzureichend gelehrt
 - Modellierung, Meta-Modellierung
 - Aktuell: Programmiersprache als Modell
 - Wenn Modelle, dann höchstens Statik-Modelle (Class-Diagram, ER-Diagram, ...)
 - Dynamik???
 - Dynamik-Modell als andere Darstellung von Code misbraucht
 - Professoren oft auch nicht geschult auf Dynamik modellen

Integration / Migration

- Integration
 - Formale Modelle von Schnittstellen
 - “Höherwertiger”: Protokollautomaten, -sprachen, QoS Informationen, Ontologien
 - Teilweise Interoperabilitätsprobleme im Generator erkennen und lösen
- Migration
 - Meta-Modelle für Migrationsschritte nötig
 - Von ... nach
 - Wann?
 - Was? Daten oder Funktionalität?
 - Definiert auf Architekturmodellen von Start und Zielarchitektur
 - Für viele Migrationsaufgaben UML2 schlecht geeignet