

# Software-Entwicklungswerkzeuge (SEW)

1

Prof. Dr. Uwe Aßmann  
Technische Universität Dresden  
Institut für Software- und  
Multimediatechnik  
<http://st.inf.tu-dresden.de>  
WS 13/14-1.0, 10/14/13

- Contents
- Goals

# Central Theses of the Course: Tools and the Engineering of Technical Spaces

2

Technical Space Bridges are important for heterogeneous software development

Technical Spaces do have the same components and tools

Model mappings for Quality Management

Generic Tools (Abstract Interpretation, logics for analysis, GRS for transformation)

Tool Construction and -Generation

Metamodels type models and programs

# Engineering of Technical Spaces

3

- ▶ Engineering of Technical Spaces is one of the most important topics of the future of software development
- ▶ Dresden has modern technologies and tools
  - Description logics
  - Invasive composition
  - Metacomposition tools (Reuseware)
  - Round-Trip Engineering and Role-based tools

# Grobgliederung der SEW-Vorlesung

4

## Teil I: Grundlagen

- ▶ Einführung zu Werkzeugen, Werkzeugmaschinen und Software-Entwicklungsumgebungen (SEU)
- ▶ Basiskonzepte in Werkzeugen
- ▶ Sprachfamilien in Werkzeugen (Struktur von M2)
- ▶ Grundlagen der Architektur von Werkzeugen
  - Metamodellgesteuerte Repositorien

## Teil II: Elementare Analysewerkzeuge

- ▶ Parsergeneratoren
- ▶ Programmanalyse
  - Abstrakte Interpretation
  - Model Checking
  - Graphanalyse

## Teil III: Architektur von Werkzeugen

- ▶ Datenintegration und Austauschformate
- ▶ Komposition von Werkzeugen
- ▶ Sprachkomposition und Werkzeugkomposition

## Teil VI: Analysewerkzeuge II

- ▶ Analyse durch Grapherreichbarkeit für Model mapping; Kantenadditionssysteme
- ▶ Attributanalyse für Visualisierung
- ▶ Interprozedurale Analyse

## Teil V: Werkzeuge zur Programmtransformation

- ▶ Werkzeuge zur Transformation; Graphtransformation
- ▶ Werkzeuge zur Optimierung
- ▶ Werkzeuge zur Programmüberführung (Codegenerierung)
- ▶ Dokumentationswerkzeuge
- ▶ MDA-Werkzeuge

## Teil VI: Werkzeuge im Software-Lebenszyklus

- ▶ Werkzeuge zur Anforderungsanalyse
- ▶ Testwerkzeuge
- ▶ Werkzeuge für Baumanagement
- ▶ Werkzeuge für Wartung und Reengineering
- ▶ Erfahrungen und weitere Entwicklungen

## Teil VII: Fortgeschrittene Themen



# Gliederung Teil I+II - Grundlagen

5

## Teil I: Grundlagen

- ▶ 10. Taxonomie von Werkzeugen u. Software-Entwicklungsumgebungen (SEU)
  - 1 Aufbau u. prinzipielle Funktion von Software-Entwicklungswerkzeugen
  - 2 Werkzeuggrundtypen - Klassen von CASE-Tools
  - 3 Werkzeug-Landschaft nach Hesse
  - 4 Einführung in Effektkategorien (Blutgruppen)
  - 5 Graph-Logik-Isomorphismus
- ▶ 11. Basistechniken: Metamodellierung und Technologieräume
  - Lifting of M2 models

- ▶ 12. Sprachfamilien in Werkzeugen
  - DDL: RM, ERD als isomorphes RM, EBNF, CD
  - DQL: Xcerpt, Semmle .QL
  - DCL: OCL, Spider, URML
  - DML: Term- und Graphersetzung
  - DRL: Data Restructuring Languages
  - BSL: DFD in SA, Mashups
- ▶ 13. Einführung in die Werkzeugarchitektur
- ▶ Teil II: Grammarware
  - 20 Parsergeneratoren
  - 22 Abstract interpretation
  - 23 Behavioral model checking of automata-based BSL
    - Standard model checking
    - Real-time model checking
  - 24. Software model checking

# Gliederung Teil III+IV – Architektur von Werkzeugen

6

## Teil III: Architektur von Werkzeugen

- ▶ 30. Architektur von Software-Entwicklungsumgebungen (SEU)
  - Konzepte der Werkzeugintegration
  - Architektur von SEU
  - Das ECMA Referenzmodell
  - Syntax-Mapping und CASE-Austauschformate
  - Frameworks zur Werkzeugintegration (PCTE)
- ▶ 31. Komposition von repository-basierten Werkzeugen
  - Effektkategorien und Blutgruppen
  - Werkzeuge als Objekte
  - Einsatz des Graph-Logik-Isomorphismus
  - Essentielle Zerlegung (EAI)
  - Repository-Integration mit Rollen
- ▶ 32. Rollenbasierte Komposition von Werkzeugen
- ▶ 33. Komposition von strombasierten Werkzeugen

## ▶ Teil IV: Meta-CASE-Umgebungen

- 40 Syntaxmapping mit EMFText
  - Für Syntaxmapping, Technologieraumbrücken und DSL-Entwicklung
- 41. MetaCase-Werkzeuge
  - MetaEdit+
- 42. FUJABA
- 43. MOFLON
- 44 Werkzeuge zur Konstruktion domänenspezifischer Sprachen
  - Reuseware zur Komposition von DSL-Fragmenten

# Gliederung Teil V+VI

7

## Teil V: Werkzeuge zur Analyse

- ▶ 50. Attributanalysen
  - Visualisierungswerkzeuge
  - VCG, AiSee, LogLin und CroCoCosmos
- ▶ 51 Werkzeuge zur Erreichbarkeitsanalyse
  - Data-flow analysis
- ▶ 52 Interprocedural Data-Flow Analysis with IDFS
- ▶ 53 Interprocedural Analysis with PAG
- ▶ 54 Worst-Case Execution Time Analysis (WCETA) mit AiT
- ▶ 55 Interprocedural Analysis with IDFS
- ▶ 55 Interprocedural Control-Flow Analysis with Graph Reachability

## Teil VI: Werkzeuge zur Transformation

- ▶ 61. Modelmanagement in Technikräumen
- ▶ 63. Werkzeuge zur Codegenerierung (Programmüberführung) und Round-Trip Engineering
  - Codeumwandlung
  - Codeergänzung und -selektion
  - Codemodifikation und -rückführung
- ▶ optional: 64 Werkzeuge für die MDA

# Gliederung Teil V+VI

8

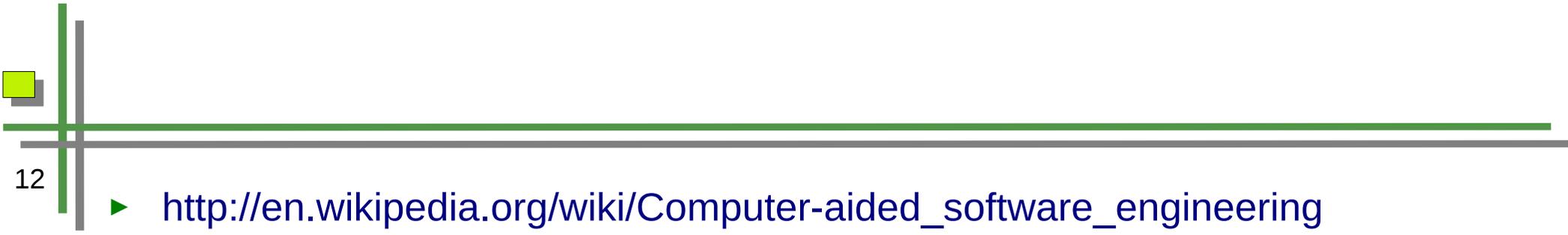
## Teil VII: Werkzeuge im Software-Lebenszyklus

- ▶ 70. Werkzeuge zur Anforderungsanalyse
  - Werkzeuggestützte Anforderungsanalyse
  - Prüf-, Dokumentations- und Verwaltungswerkzeuge
- ▶ 71. Testwerkzeuge
  - Aufgaben und Arten von Testwerkzeugen
  - Qualitätsmanagement auf der Basis von Modellabbildungen
  - Werkzeuge ausgewählter Testumgebungen
  - Simulation: Debugger
- ▶ opt. 72 Dokumentationswerkzeuge
  - Werkzeuge zum Konfigurationsmanagement werden in Kurs “Softwaremanagement” behandelt

- ▶ 73. Baumanagement
  - Das Baumanagementsystem von GNU
  - Fehlermanagement
- ▶ 74. Werkzeuge für Wartung und Reengineering
  - Reengineering, Modellabbildung und QM
  - Werkzeuge für das Reengineering
  -
- ▶ Teil VIII: Fortgeschrittene Themen
  - Hybride MDSD
  - Simulationssprachen

- ▶ Verstehe, wie Technikräume die Entwicklung von Software beeinflussen (Multi-technical space development)
- ▶ Verstehe, wie Software-Werkzeuge in einem Technikraum funktionieren
  - Architektur
  - Komponenten
  - Sprachen
  - Technikräume, Modellalgebren und Metasprachen
- ▶ Verstehe, wie man sie zusammensteckt (komponiert) unter Komposition der Metamodelle und Sprachen
- ▶ Verstehe Software-Entwicklungsumgebungen als zusammengesteckte Werkzeuge
- ▶ Lerne wichtige Sprachen kennen, die Werkzeuge prägen
  - mit denen man einzelne Werkzeuge bauen kann
- ▶ Unterscheide code- und modellzentrierte Werkzeuge
- ▶ Verstehe Werkzeuge für den Software-Entwicklungsprozess

- [1 B93] Balzert, H. (Hrg.) u.a.: CASE - Systeme und Werkzeuge; BI-Wissenschaftsverlag Mannheim, 5. vollst. überarb. Auflage 1993
- [2 BAL, BOO] Balzert, H. : Lehrbuch der Software-Technik (Bd. I): Software-Entwicklung; Spektrum Akademischer Verlag (3. Auflage) Heidelberg 2009
- [3 BAL-II B98] Balzert, H. : Lehrbuch der Software-Technik (Bd. II): Software-Management. Spektrum Akademischer Verlag Heidelberg 1998
- [4 HMF] Hesse, W., Merbeth, G., Fröhlich, R.: Software-Entwicklung / Vorgehensmodelle. Projektführung, Produktverwaltung; Oldenbourg Verlag München 1992
- [5 SN92] Schönthaler, F., Nemeth, T.: Software-Entwicklungswerkzeuge: Methodische Grundlagen; B.G. Teubner Verlag Stuttgart 1992
- [6 ES89] Engels, G., Schäfer, W.: Programmmentwicklungsumgebungen - Konzepte und Realisierung; B.G. Teubner Verlag Stuttgart 1989
- [7 ZK04] Zeller, A., Krinke, J.: Open-Source-Programmierwerkzeuge (2. Auflage) dpunkt.verlag Heidelberg 2004
- [8 F91] Fisher, A.S.: CASE - Using Software Development Tools (Second Edition); John Wiley & Sons 1991
- [9 B92] Bauknecht, K. (Hrg.): Informatik-Anwendungsentwicklung - Praxiserfahrungen mit CASE; B.G. Teubner Verlag Stuttgart 1992
- [Züll] Züllighoven, Heinz. Object-Oriented Construction Handbook; dpunkt.verlag 2005
- [Raasch] Raasch. Systementwicklung mit strukturierten Methoden. Hanser. 1993



▶ [http://en.wikipedia.org/wiki/Computer-aided\\_software\\_engineering](http://en.wikipedia.org/wiki/Computer-aided_software_engineering)

