

Softwaretechnik

Model Driven Architecture

Introduction

Prof. Dr. Peter Thiemann

Universität Freiburg

20.07.2011

PT (Univ. Freiburg)

Softwaretechnik Model Driven Architecture Introduction
Introduction MDA

20.07.2011

1 / 15



What is MDA?

- ▶ MDA = Model Driven Architecture
 - ▶ also: MD (Software/Application) Development, Model Based [Development/Management/Programming]
 - ▶ Model Driven Engineering, Model Integrated Computing
- ▶ Initiative of the OMG (trade mark)
 - ▶ OMG = Object Management Group: CORBA, UML, ...
 - ▶ open consortium of companies (ca. 800 Firmen)
- ▶ Goal: Improvement of software development process
 - ▶ Interoperability
 - ▶ Portability
- ▶ Approach: Shift development process from code-centric to model-centric
 - ▶ Reuse of models
 - ▶ Transformation of models
 - ▶ Code generation from models

PT (Univ. Freiburg)

Softwaretechnik Model Driven Architecture Introduction

20.07.2011

3 / 15



Introduction MDA

Material

- ▶ Thomas Stahl, Markus Völter. Model-Driven Software Development. Wiley & Sons. 2006.



- ▶ Anneke Kleppe, Jos Warmer. MDA Explained: The Model Driven Architecture: Practice and Promise. Pearson. 2003.
- ▶ Stephen J. Mellor, Axel Uhl, Kendall Scott, Dirk Weise. MDA Distilled: Solving the Integration Problem with the Model Driven Architecture. Pearson. 2004.

PT (Univ. Freiburg)

Softwaretechnik Model Driven Architecture Introduction
Introduction MDA Goals of MDA

20.07.2011

2 / 15



Goals of MDA

Higher Degree of Abstraction

Portability and Reusability

- ▶ Development abstracts from target platform
- ▶ Technology mapping in reusable transformations
- ▶ New technology \Rightarrow new transformation

Interoperability

- ▶ Systems span several platforms
- ▶ Information flow between platforms via *bridges*
- ▶ Byproduct of model transformations

PT (Univ. Freiburg)

Softwaretechnik Model Driven Architecture Introduction

20.07.2011

4 / 15



Goals of MDA

Models and Model Transformations

Productivity

Every development phase directly contributes to the product, not just the implementation

Documentation and Maintenance

- ▶ Changes through changes of the models
- ▶ Models are documentation \Rightarrow consistency
- ▶ Separation of concern
- ▶ Better handle on changing technology

Specialization

- ▶ Business processes
- ▶ Technologies

The Concept "Model"

(according to Herbert Stachowiak, 1973)

Representation

A model is a representation of an original object.

Abstraction

A model need not encompass all features of the original object.

Pragmatism

A model is always goal-oriented.

- ▶ Modeling creates a representation that only encompasses the relevant features for a particular purpose.

The Concept "Model"

(according to Herbert Stachowiak, 1973)

Representation

A model is a representation of an original object.

Abstraction

A model need not encompass all features of the original object.

Pragmatism

A model is always goal-oriented.

Formal Models

Models authored in a formal language

- ▶ Textual: defined by grammar, BNF, etc
- ▶ Grafical: defined by *Metamodel*
 - ▶ Which modeling elements?
 - ▶ Which combinations?
 - ▶ Which modifications?

Models with a formal semantics

- ▶ Example: logical formula \Rightarrow truth value
- ▶ Example: context-free grammar \Rightarrow language
- ▶ Example: program \Rightarrow programm execution

Why Formal Models?

Model Editor

- ▶ manipulation of models
- ▶ requires formal definition

Why Formal Models?

Model Editor

- ▶ manipulation of models
- ▶ requires formal definition

Model Transformation

- ▶ transforming a model in one or more target models
- ▶ requires formal definition / formal semantics

Model Verification

- ▶ properties: interface, timing, ...
- ▶ relation between model and original
- ▶ requires formal definition and formal semantics

Why Formal Models?

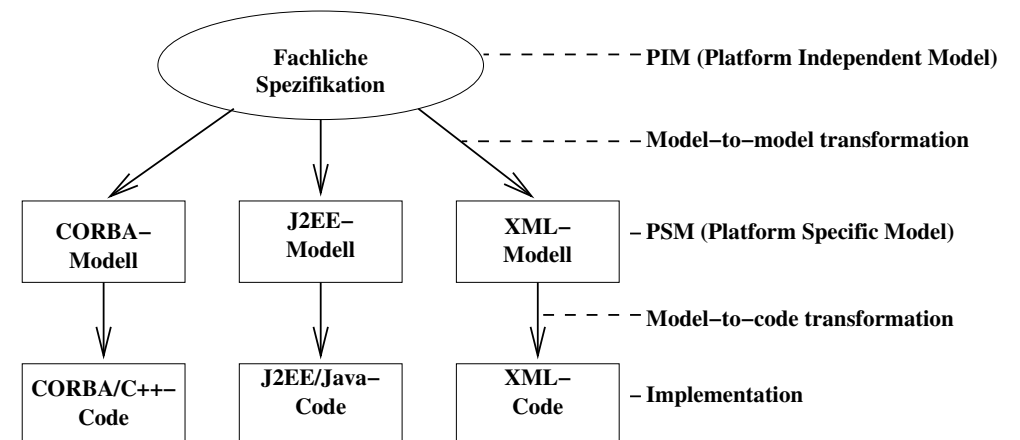
Model Editor

- ▶ manipulation of models
- ▶ requires formal definition

Model Transformation

- ▶ transforming a model in one or more target models
- ▶ requires formal definition / formal semantics

Models in MDA



Models in MDA/2

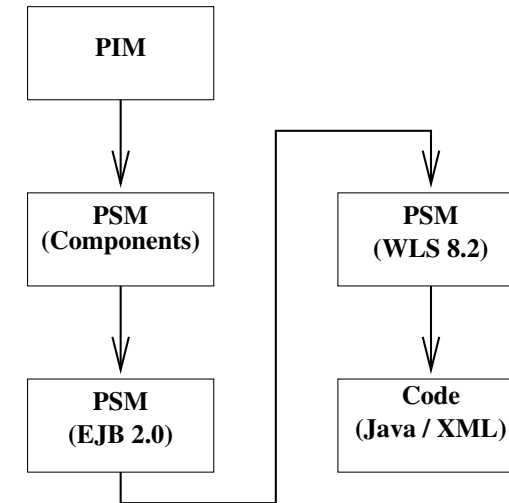
PIM vs PSM

- ▶ Relative concepts
- ▶ Smooth transition
- ▶ Several levels of model and transformation steps possible
- ▶ Inverse transformation PSM \Rightarrow PIM unlikely

Transformation

- ▶ Code is the ultimate model (PSM)
- ▶ Model-to-code is a special case

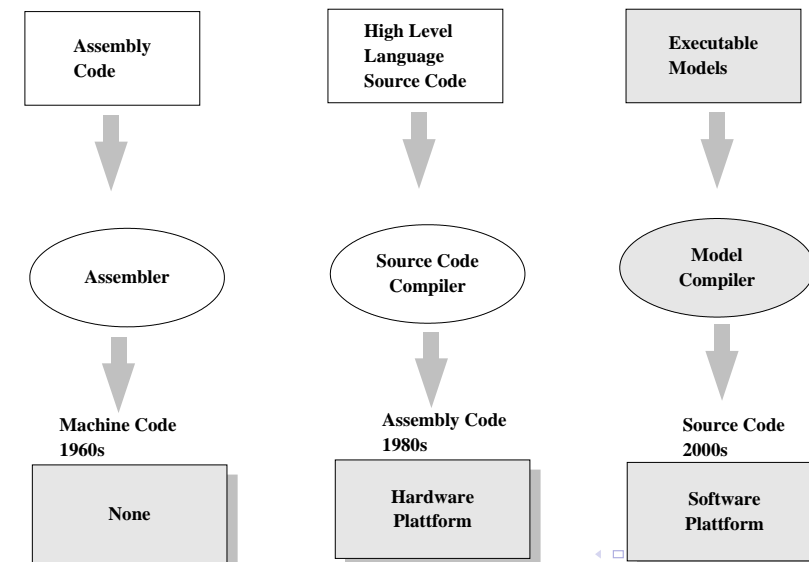
Models and Transformations



Platform

- ▶ API
- ▶ Virtual machine
- ▶ Provides several services
- ▶ Examples
 - ▶ Different processors \Rightarrow hardware platform
 - ▶ Operating system \Rightarrow software platform
 - ▶ Java VM \Rightarrow software platform
 - ▶ EJB \Rightarrow component platform
 - ▶ CORBA, Webservices, ...
 - ▶ Application architecture, DSL (Domain Specific Language)

Examples for Platforms



Transformations

- ▶ Mappings between models
- ▶ Formal definition required for automatic application
- ▶ Standardized transformation language QVT (Queried Views and Transformations)
Distilled from 23 very different proposals
- ▶ Tools
 - ▶ Transformations based on metamodel
 - ▶ Code generator via patterns
 - ▶ Proprietary transformation languages (scripting)
- ▶ Currently lack of interoperability
4 implementations of parts of the standard

Next Steps

- ▶ Metamodeling