

64. Tools for Model-Driven Architecture (MDA) (Werkzeuge für die modellgetriebene Architektur)

Prof. Dr. rer. nat. Uwe
Aßmann

Institut für Software- und
Multimediatechnik

Lehrstuhl Softwaretechnologie

Fakultät für Informatik

TU Dresden

<http://st.inf.tu-dresden.de>

Version 13-0.4, 01.01.14

- 1) MDA
- 2) Model Mapping
- 3) Model Transformation
- 4) Model2Code Transformation

1) QVT

1) ATL



SEW, © Prof. Uwe Aßmann

1

Literature

- 2 ▲ Alan Brown. An introduction to Model Driven Architecture. Part I: MDA and today's systems
 - <http://www.ibm.com/developerworks/rational/library/3100.html>
- ▲ Frédéric Jouault and Ivan Kurtev. On the Architectural Alignment of ATL and QVT. In: Proceedings of the 2006 ACM Symposium on Applied Computing (SAC 06). ACM Press, Dijon, France, chapter Model transformation (MT 2006), pages 1188—1195.
 - <http://atlanmod.emn.fr/bibliography/SAC06a>
- ▲ Tutorial über ATL “Families2Persons”
 - http://www.eclipse.org/m2m/atl/doc/ATLUseCase_Families2Persons.ppt
- ▲ ATL Zoo von Beispielen
 - <http://www.eclipse.org/m2m/atl/atlTransformations>
- ▲ Kevin Lano. Catalogue of Model Transformations
 - <http://www.dcs.kcl.ac.uk/staff/kcl/tcat.pdf>
- ▲ Implementation in ATL
 - <http://www.eclipse.org/m2m/atl/atlTransformations/EquivalenceAttributesAssociations/EquivalenceAttributesAssociations.pdf>



Model-Driven Software Development (MDSD)

3

- ▶ MDSD falls into two main development methods:
 - Engineering with DSL and Metamodels (domain-specific modeling, DSM)
 - Model-Driven Architecture (MDA)
- ▶ **Model mappings** correlate models defining equivalence relations between model elements
 - From them, model transformations can easily be derived
- ▶ **Model transformations**
 - **Horizontal model transformations** transform a model within a single language
 - **Vertical model transformations** transform a model from a higher-level language to a lower-level language (**lowering**)
 - **Broadband model transformations** transform a model from a higher-level set into a lower-level set of a broadband (wide-spectrum) language
- ▶ **Model weavings** extend models by other models

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)



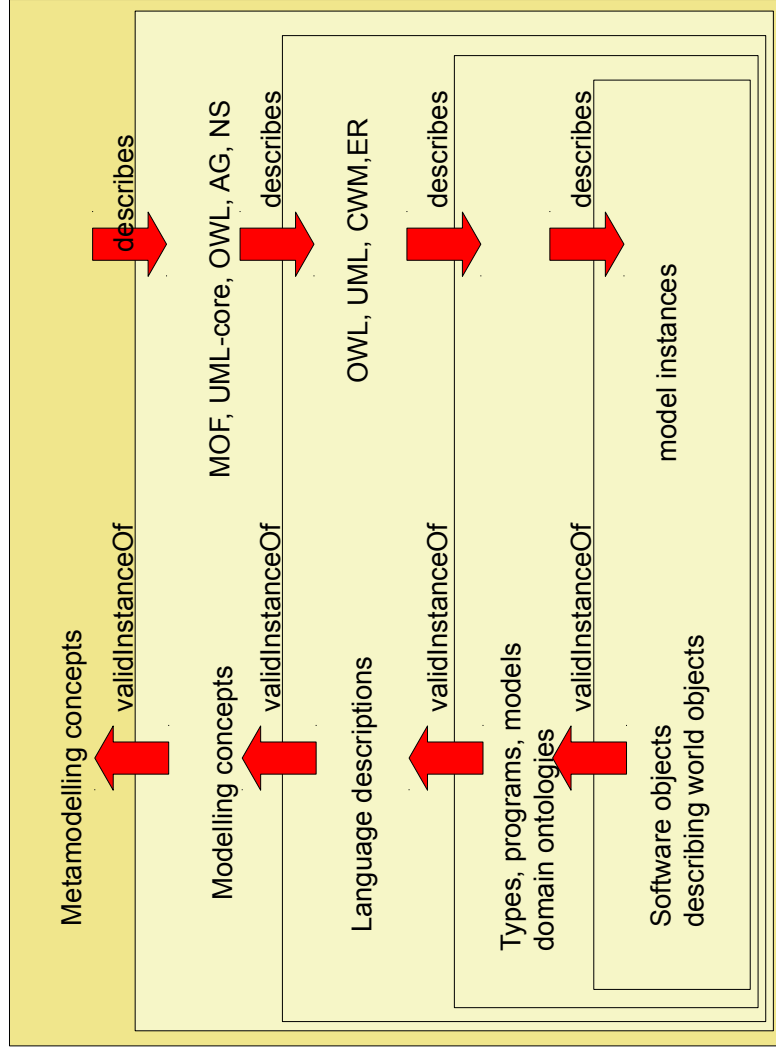
64.1 Modellgetriebene Architektur (MDA)



The IRDS/MOF Metamodelling Hierarchy

5

▶ aka *metapyramid*



M4 level = M3

M3 metamodel level

M2 metamodel level

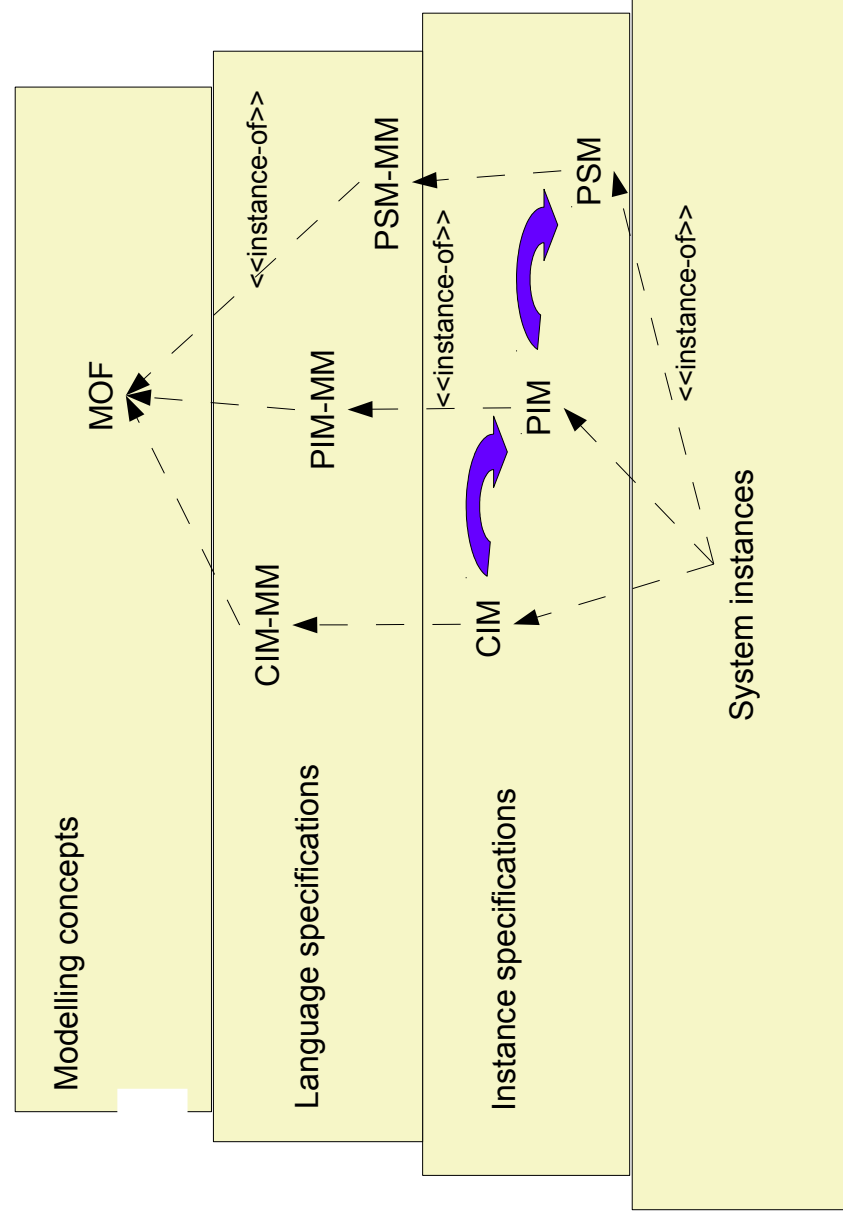
M1 model level

M0 Object level



The MDA Embedded in the IRDS Metapyramid

6



M3 metamodel level

Metametamodels

M2 metamodel level
Metamodels (Languages)

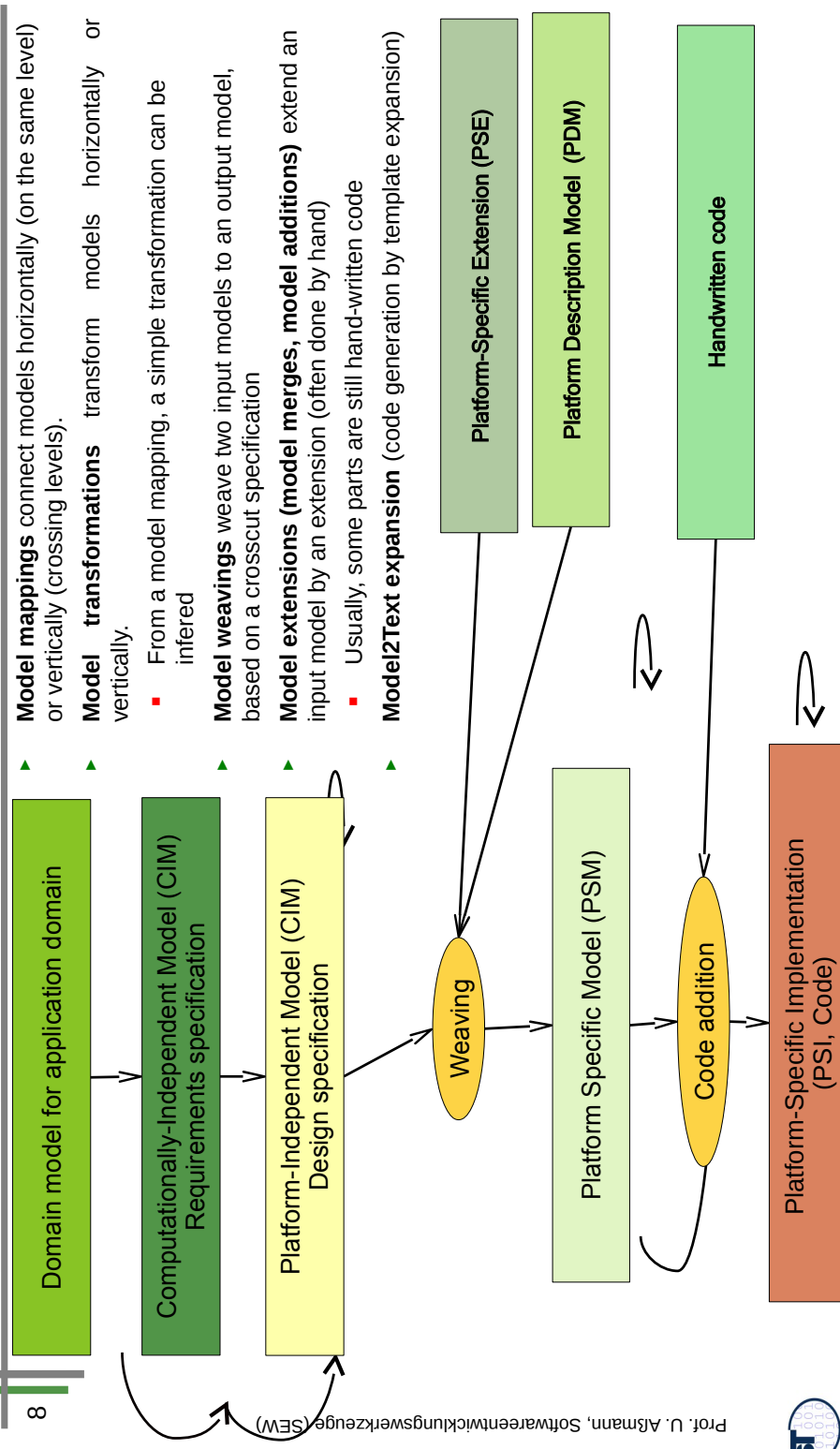
M1 model level
Models

M0 object level



- ▶ Modelle und Spezifikationen der MDSD und der MDA werden in einem Werkzeug als (abstrakt Syntax-)Graphen (ASG) repräsentiert
- ▶ MDA-Werkzeuge benötigen Graph-Mapping, Graph-Querying und Graphtransformation
 - Gewöhnlich bieten sie Unterstützung für eine oder mehrere der besprochenen Sprachen
 - Typisierung durch Metamodelle
- ▶ Oft werden die Sprachen QVT oder ATL unterstützt

Model Mappings and Model Weavings



Modell-Verknüpfung (Model Mapping) am Beispiel MID INNOVATOR

- ▶ Innovator kann gleichzeitig für Analyse-, Entwurfs- und Implementierungsmodelle eingesetzt werden, sowie für Transformationen dazwischen
- ▶ Wie kann man diese Modelle systemisch mit einander verknüpfen?

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)

The screenshot shows the UML Modeler interface. On the left, a project tree displays the following structure:

- TTBib_UML
 - systemModel
 - external object
 - Use Case System
 - analysis system
 - Java design system
 - Java implementation system \$IHOTMP/src
 - systemModel management

On the right, a table lists model elements with their status, names, types, and change dates:

Status	Name	Typ	Änderungsdatum
1 0 A	Ausleihe	Sec...	22.11.2003 00:48:02
2 0 A	Kunde_anmelden	Koll...	10.11.2003 01:21:54
3 0 A	Rückgabe	Sec...	22.11.2003 00:21:47
4 0 A	Tonträger_Einkauf	Sec...	10.11.2003 01:23:59
5 0 A	Kunden_neu_anlegen	Sec...	10.11.2003 01:26:19
6 0 A	AnalysisClassDiagram	Klas...	09.11.2003 15:29:14
7 0 A	Verwaltung_AS	Klas...	09.11.2003 15:25:56
8 0 A	Tonträger_AS	Klas...	09.11.2003 15:20:08
9 0 A	Kunde_AS	Klas...	09.11.2003 15:27:32
... 0 A	: Kunde_AS	Obji...	09.11.2003 13:20:05
... 0 A	: Tonträger_AS	Obji...	09.11.2003 13:20:16
... 0 A	VerwaltungUl_AS	Klas...	09.11.2003 15:16:32
... 0 A	: VerwaltungUl_AS	Obji...	09.11.2003 13:23:08
... 0 A	: Kunde_LUC	Obji...	09.11.2003 14:05:54
... 0 A	Bibliothek_LUC	Obji...	09.11.2003 15:44:35
... 0 A	: Verwaltung_AS	Obji...	09.11.2003 16:14:14



PIM und PSM gemäß der MDA

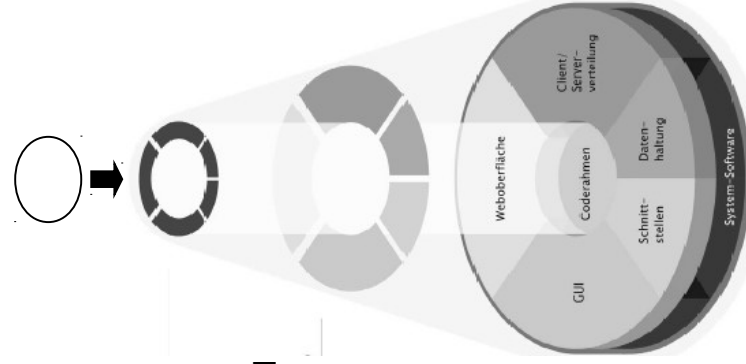
10 Für die unterschiedlichen Abstraktionsebenen **PIM** und **PSM** stehen verschiedene Beschreibungs- mittel zur Verfügung:

Fachkonzept auch CIM
(Computation independent model)

Plattformunabhängiges Modell
(UML, OCL, XMI)

Plattformspezifisches Modell PSM
Basiskomponenten (JB)
Steuerungskomponenten
Infrastrukturkomponenten (EJB,
CCM, COM+, .NET)
Anwendungskomponenten

Plattformspezifische Implementierung (PSI)
in Programmiersprache

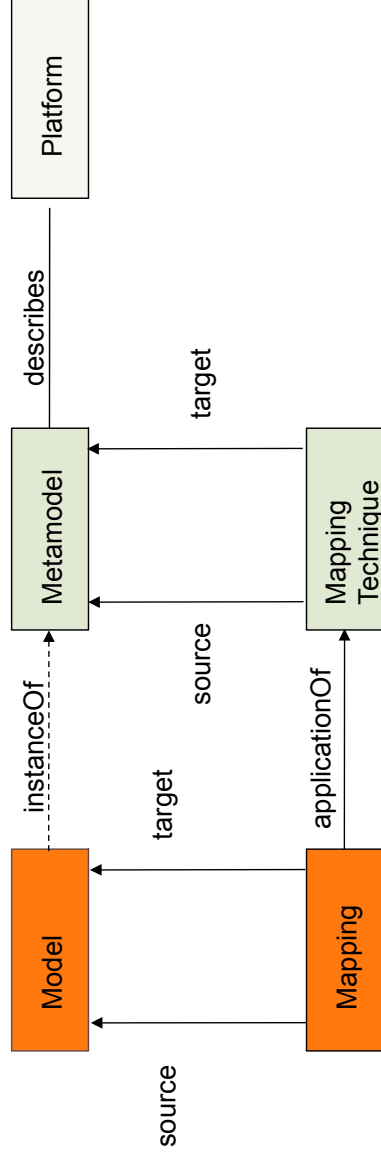


Ein **PSM** berücksichtigt die jeweilige Basistechnologie, auf der ein **PIM** zum Einsatz kommen kann (CORBA-Broker, .NET-Spezifika oder das Web-Service-Protokoll SOAP).
Auch **PSMs** können mit der UML modelliert werden. In jedem Fall werden aus den **PSMs** die **Codegerüste** erzeugt, die die Komponenten-Entwickler dann weiter bearbeiten.



What are Model Mappings?

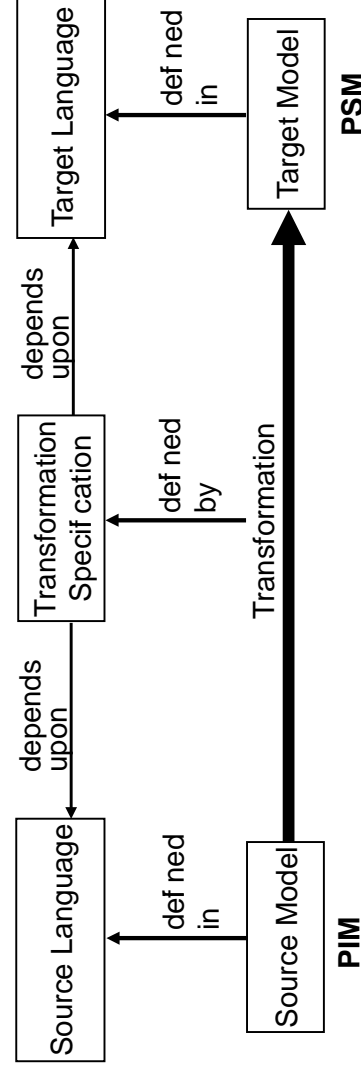
- Remember Model: "A model is a representation of a part of a function of a system, its structure, or behavior"
- A model mapping can be generated from a model analysis
- The mappings are automatic or semi-automatic: step-wise refinement of the model by transformation
 - From a model mapping, transformations can be generated



MDA-Transformationsprozess

Aus plattformunabhängigem (*independent*) Metamodell **PIM** sind mittels Regeln, Techniken platfformspezifische (*specific*) Modelle **PSM** zu entwerfen, zu generieren, oder abzuleiten, um neue Anwendungen für eine bestimmte (Komponenten-)Plattform zu erhalten.

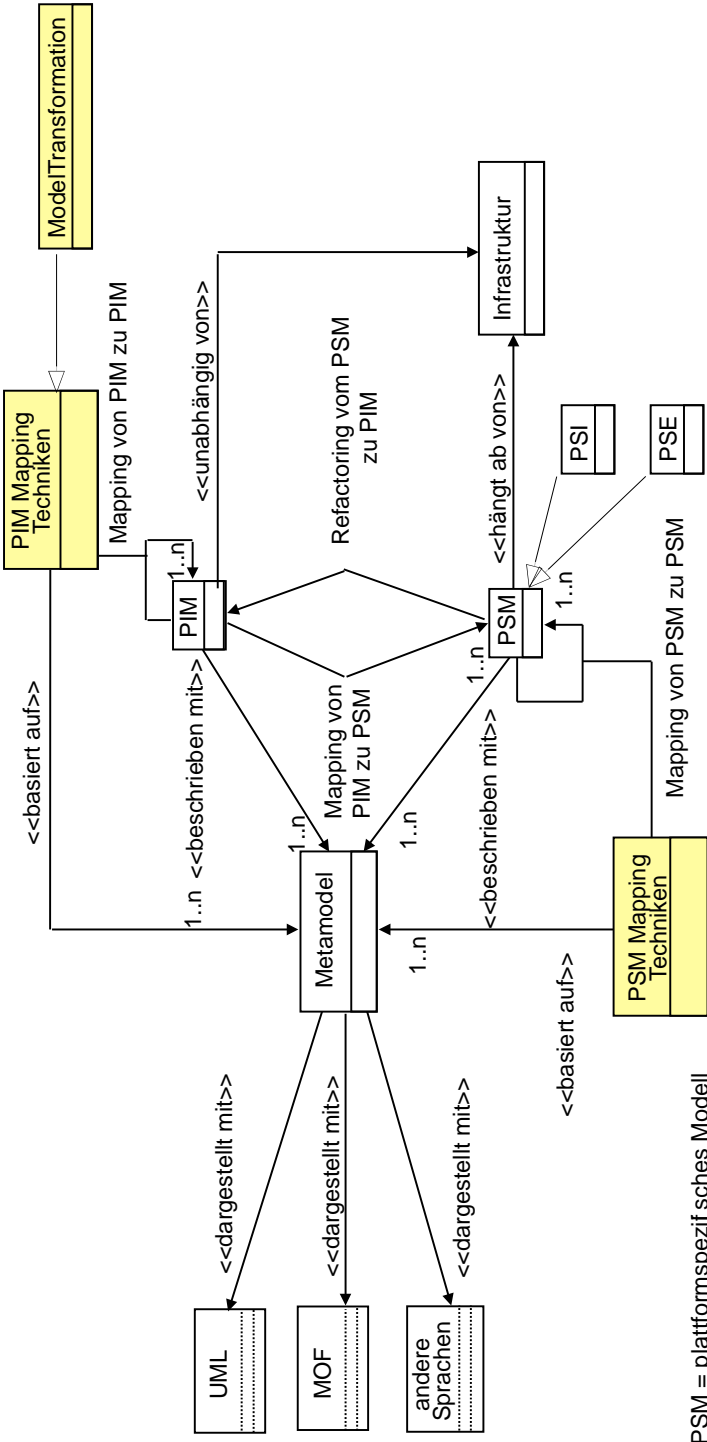
Ein weiteres Ziel von MDA ist die Integration solcher Technologien wie CORBA, J2EE, .Net und XML als *Plattform*.



Quelle: Kleppe, A., Warmer, J., Bast, W.: MDA Explained - Practice and Promise of the Model Driven Architecture; Addison Wesley 2003 (Draft 25.10.02)

Das MDA-Metamodell

13



Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)

Transformationen bezeichnet man auch als **Abbildungen (mappings)**. Mapping von PIM zu PIM schafft neue „Business Viewpoints“, von PSM zu PIM Abstraktionen aus plattformabhängigen Implementierungen und zwischen PSM weiteren Verfeinerungen oder Zielpattformen.



Model Management

14

- ▶ In der MDA müssen Modelle (Graphen) verwaltet werden:
 - Modellalgebren
 - Lookup
 - Diff, comm, union, compose
 - Versionsmanagement
 - Konfigurationsmanagement
- ▶ Das führt auf metamodelgesteuerte Repositorien/Modellinfrastrukturen (siehe Kapitel “Repositories” und “Modelmanagement”)

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)



Bewertungsaspekte von MDA-Tools

15

- ▶ Unterstützung der **Metamodellierung**
 - Metamodelle der Sprachen UML 2.0, OCL, CWM (MOF 2.0-basiert)
 - Metamodellgesteuerte Repositorien
 - Erweiterungsmöglichkeiten der UML-Profile durch explizite Metamodellierung sowie Modellprüfung
 - Austauschformate: Import, Export und Validierung von Modellen auf Basis ihres Austauschs mit XML 2.0
 - Validierung der Modelle mit OCL
- ▶ **Model-to-Model Mapping** bzw. **Model-to-Model Transformation** (z. B. PIM zu PSM) mit QVT, ATL, Graphersetzung oder einer proprietären Sprache
- ▶ **Forward- und Reverse-Engineering**
 - Codegenerierung (Model-to-Code Transformation, PSM zu PSI)
 - Mapping zu einer Programmiersprache wie z. B. JMI
- ▶ Roundtrip-Engineering auf der Code-Ebene zur Unterstützung des Single-Source-Prinzips
- ▶ Modellierung von Testfällen und automatische Generierung der Testdaten (Model-driven Testing)

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)



Quelle: Petrasch, R., Meimberg, O.: Model Driven Architecture - eine praxisorientierte Einführung in die MDA; dpunkt-verlag 2006

Werkzeugfunktionen am Bsp. ArcStyler

16

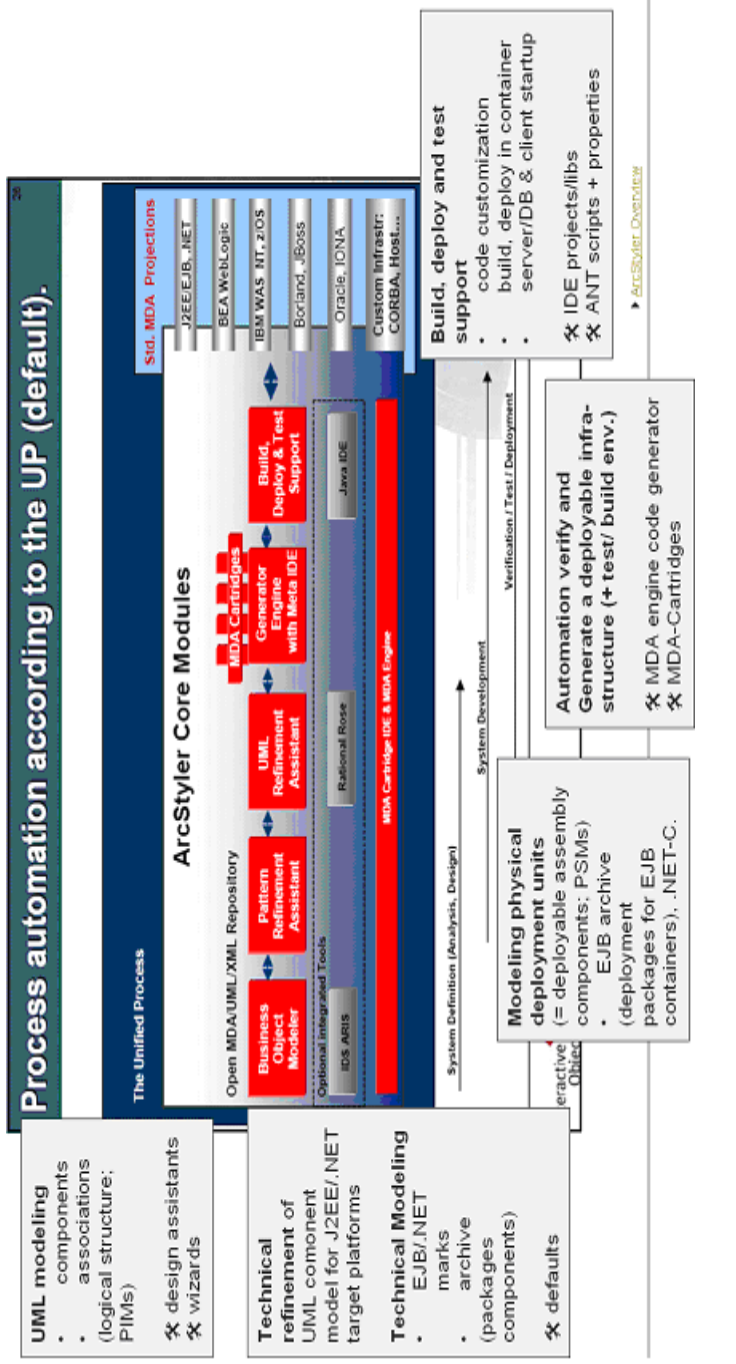
- Das Werkzeug ArcStyler ist im Zusammenspiel mit einem UML-Editor wie MagicDraw (oder Rational Rose...) ein leistungsfähiges Werkzeugsystem (MDA-Suite), mit dem sich zum Beispiel J2EE-Applikationen gemäß den Konzepten der MDA entwickeln lassen.
- ▶ Object Modeler erfasst Anforderungen unabhängig von Plattform (funktionale, Anforderungen) Basis CRC-Cards Technologie
 - ▶ Pattern Refinement Assistent überführt Fachmodell interaktiv in PIM UML-Modell (Basis MagicDraw oder Rational Rose) mit Annotation der essentiellen Design-Entscheidungen
 - ▶ Verfeinerung des Fachmodells top down in untergeordnete UML-Diagramme und Quellcodegenerierung ebenfalls mit UML-Tool (MagicDraw)
 - ▶ Codevollständigung und Optimierung für jeweiligen Applikationsserver mit Cartridges (Codegenerierungs-Plugins)
 - ▶ Komponentengenerierung für Oberfläche sowie weitere Projekt- und Konfigurationsdateien mit JBuilder.
 - ▶ Schnittstelle zu IDE ist Standard „Ant Build Process“
 - ▶ Datenbankgenerierung über Skripte zum Erstellen der DB-Schemas möglich.

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)



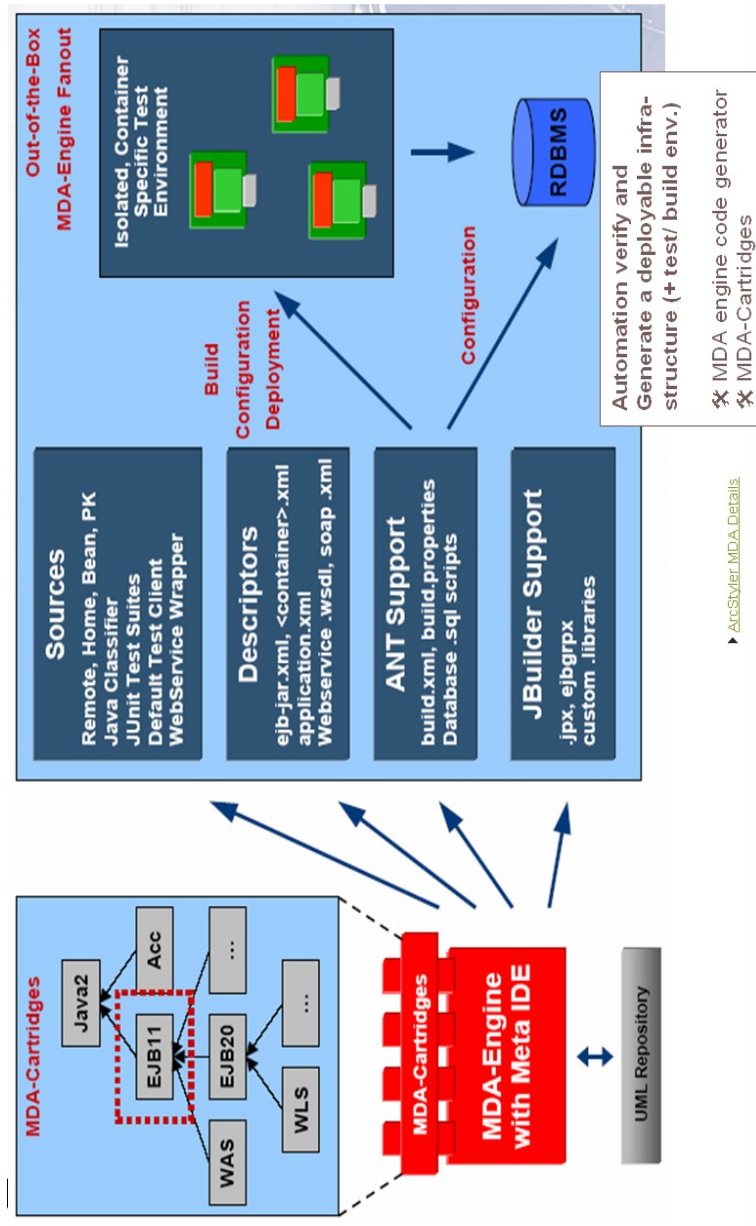
Quelle: Versteegen, G.: Wege aus der Plattformabhängigkeit - Hoffnungsträger Model Driven Architecture; Computerwoche 29(2002) Nr. 5 vom 1. Febr. 2002

Vorgehen und Unterstützung beim ArcStyler



<http://www.interactive-objects.com/products/arcstyler/supportdocumentation.html>

Cartridges und generierte Artefakte



Quelle: Butze, D.: Entwicklung eines Praktikums für die werkzeuggestützte Softwareentwicklung nach der Model-Driven-Architecture; Großer Beleg an der Fakultät Informatik der TU Dresden 2004

Kurzbeschreibung weiterer MDA-Tools

19

	Integriert in	URL
AndroMDA	Eclipse	http://www.andromda.org/
XText, Xpand	Eclipse	http://www.eclipse.org/Xtext/
IBM Rational Suite Software Architect	Eclipse	
BITplan smart Generator	Eclipse	http://www.bitplan.com/

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)



Quelle: Petrasch, R., Meimberg, O.: Model Driven Architecture - eine praxisorientierte Einführung in die MDA; dpunkt-verlag 2006

End

20

Prof. U. Asmann, Softwareentwicklungswerkzeuge (SEW)

