

Architektura Sterowana Modelem *Model Driven Architecture*

prezentacja 1

Wprowadzenie

wersja 1.0

dr inż. Paweł Głuchowski

Wydział Informatyki i Telekomunikacji, Politechnika Wrocławska

Treść prezentacji

1. Co to jest Architektura Sterowana Modelem (MDA)
2. Plan wykładów
3. Spis prezentacji
4. Pomocne książki i strony internetowe

1

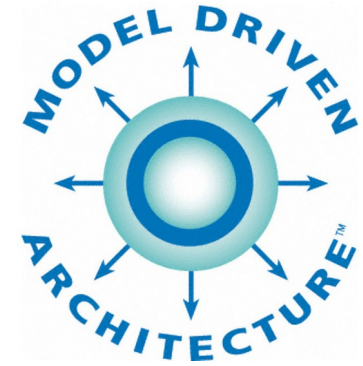
Co to jest Architektura Sterowana Modelem (MDA)

Co to jest Architektura Sterowana Modelem (MDA)

Architektura Sterowana Modelem

/Model Driven Architecture, MDA/

na podst. [MDA - The Architecture Of Choice For A Changing World](#)



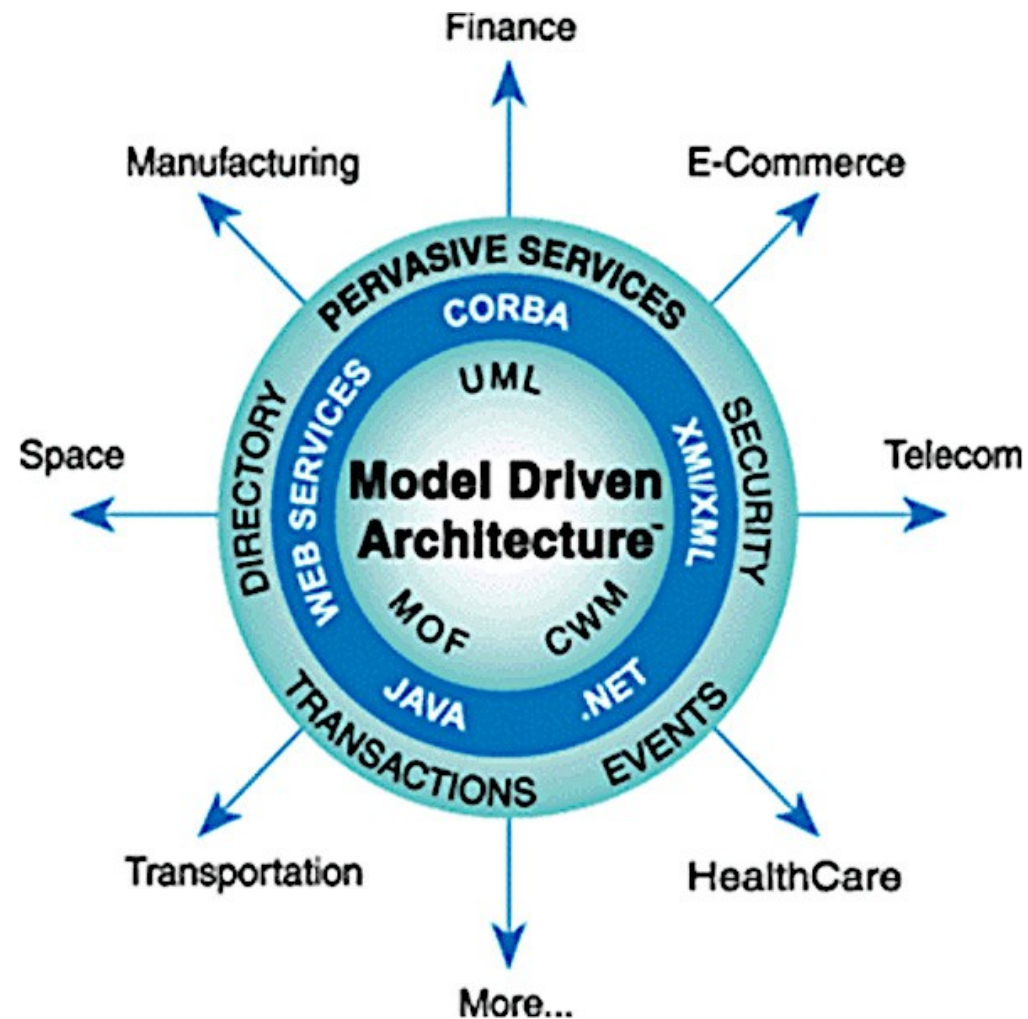
- Metodyczne podejście do inżynierii oprogramowania (projektowania, rozwijania i implementowania oprogramowania).
 - Oparte na budowie modeli (abstrakcji systemu) i ich transformacji.
 - Opracowane przez *OMG Standards Development Organization*.
- Wytyczne strukturyzacji specyfikacji oprogramowania (w tym systemu zintegrowanego)
 - przy pomocy modeli niezależnych od platformy programistycznej (w tym Web Services, .NET, CORBA R, J2EE)
- Graficzne modelowanie funkcjonalności i biznesowego zachowania:
 - głównie w języku UML,
 - niezależnie od wybranej technologii ich implementacji.

Co to jest Architektura Sterowana Modelem (MDA)

Architektura Sterowana Modelem

/Model Driven Architecture, MDA/

- Izolacja rdzenia oprogramowania od techniki jego implementacji i cyklu zmian:
 - logika biznesowa pozostaje zgodna z biznesowymi wymaganiami,
 - a technologia oprogramowania może się rozwijać i zmieniać.
- Interoperacyjność w ramach i poza platformą programistyczną.
- **„Wszystko jest modelem”**
 - zamiast klasycznego podejścia do programowania, gdzie *„wszystko jest obiektem”*.



Architektura Sterowana Modelem

/Model Driven Architecture, MDA/

- **Kolejne poziomy abstrakcji modelu**
(od koncepcji do szczegółowej implementacji systemu):
 - **CIM** (*Computation-Independent Model*) – model niezależny od środków informatycznych,
 - **PIM** (*Platform-Independent Model*) – model niezależny od platformy programistycznej i systemowej,
 - **PSM** (*Platform-Specific Model*) – model specyficzny dla platformy programistycznej i systemowej.
- Na podstawie modelu PSM powstaje kod oprogramowania.

2

Plan wykładów

Spodziewane efekty

- **Z zakresu wiedzy:**
 - student zna metodę MDA w tworzeniu oprogramowania,
 - student zna język modelowania UML i jego metamodel,
 - student zna typowe wzorce projektowe i ich klasyfikację,
 - student zna metody translacji języków tekstowych i graficznych.
- **Z zakresu kompetencji społecznych:**
 - student rozumie, czym skutkuje brak precyzji i użycia standardów w modelowaniu oprogramowania w jego późniejszej implementacji i wdrożeniu;
 - student ma świadomość, jakie wsparcie dla pracy zespołowej nad projektem informatycznym daje graficzne modelowanie oprogramowania.

Spodziewane efekty

- **Z zakresu umiejętności:**
 - student potrafi wybrać i zastosować właściwe metody i narzędzia MDA w modelowaniu oprogramowania – systemu informatycznego;
 - student potrafi wykonać modele oprogramowania: CIM, PIM i PSM przy pomocy odpowiednich diagramów UML, stosując odpowiednie wzorce projektowe i narzędzia;
 - student potrafi wykonać i użyć translator modelu oprogramowania na jego inny model;
 - student potrafi wykonać i użyć translator modelu oprogramowania na jego kod.

Plan wykładów

- **Wykład 6:**
 - Projektowanie systemu informatycznego metodą MDA (Model Driven Architecture – Architektura Sterowana Modelem).
- **Wykłady 7 i 8:**
 - Język UML w projektowaniu oprogramowania metodą MDA.
- **Wykład 9:**
 - Wzorce projektowe w budowie modelu oprogramowania.
- **Wykład 10:**
 - Definiowanie metamodelu.
 - Narzędzia wspierające MDA.
- **Wykłady 11–14:**
 - Transformacja modelu w model (M2M) i modelu w tekst (M2T).
- **Wykład 15:**
 - Omówienie wybranych projektów.

3

Spis prezentacji

Spis prezentacji

1. Wprowadzenie
2. Projektowanie systemu informatycznego metodą MDA
3. Język UML – diagramy zachowań
4. Język UML – diagramy struktury
5. Język UML – modelowanie systemu informatycznego
6. Wzorce projektowe w budowie modelu oprogramowania
7. Definiowanie metamodelu
8. Narzędzia wspierające MDA
9. Transformacja M2M modelu PIM w model PSM
10. Transformacja M2T modelu PSM w tekst kodu

4

Pomocne książki i strony internetowe

Dokumentacja MDA i wspierających technologii

- **The OMG Specification Catalog**
– www.omg.org/spec
- **MDA Specifications**
– www.omg.org/mda/specs.htm
- **MDA Guide rev. 2.0**
– www.omg.org/cgi-bin/doc?ormsc/14-06-01
- **Unified Modeling Language (UML)**
– www.omg.org/spec/UML
- **MetaObject Facility (MOF)**
– www.omg.org/spec/MOF
- **XML Metadata Interchange (XMI)**
– www.omg.org/spec/XMI
- **Common Warehouse Metamodel (CWM)**
– www.omg.org/spec/CWM

Inżynieria Oprogramowania

- **Specyfikacja wymagań oprogramowania. Kluczowe praktyki analizy biznesowej**
– K. Wiegers, C. Hokanson; wyd. Helion 2024
- **Inżynieria oprogramowania w praktyce. Od wymagań do kodu z językiem UML**
– M. Śmiałek, K. Rybiński; wyd. Helion 2023
- **Inżynieria oprogramowania**
– A. Jaskiewicz; wyd. Helion 2011
- **Inżynieria oprogramowania**
– K. Sacha; wyd. PWN 2010
- **Inżynieria oprogramowania**
– J. Somerville; wyd. WNT 2003
- **Inżynieria oprogramowania w projekcie informatycznym**
– J. Górski; wyd. Mikom 1999

Inżynieria Oprogramowania

- **Inżynieria Oprogramowania, Modelowanie i analiza systemów informatycznych, Projektowanie oprogramowania**
 - prezentacje wykładowe; Z. Kruczkiewicz; PWr
 - zofia.kruczkiewicz.staff.iiar.pwr.edu.pl/index.php?id=dydaktyka
- **Inżynieria oprogramowania, Kierowanie projektem programistycznym,**
 - prezentacje wykładowe; O. Unold; PWr
 - olgierd.unold.staff.iiar.pwr.wroc.pl/dydaktyka
- **PSK - projektowanie systemów komputerowych**
 - B. Frączak; 2009
 - brasil.cel.agh.edu.pl/~09sbfraczek
- **Learning Guides**
 - Visual Paradigm
 - www.visual-paradigm.com/guide

Wzorce projektowe

- **Tablice Informatyczne. Wzorce projektowe**
– D. Krasnokucki; wyd. Helion 2017
- **Projektowanie zorientowane obiektowo. Wzorce projektowe**
– A. Shalloway, J.R. Trott; wyd. Helion 2005
- **J2EE. Wzorce projektowe**
– D. Alur, J. Crupi, D. Malks; wyd. Helion 2004
- **Wzorce projektowe**
– Refactoring.Guru
– refactoring.guru/pl/design-patterns

MDA

- **MDA - The Architecture Of Choice For A Changing World**
 - OMG
 - www.omg.org/mda
- **MDA Specifications**
 - OMG
 - www.omg.org/mda/specs.htm
- **About the MOF Model to Text Transformation Language Specification Version 1.0**
 - OMG
 - www.omg.org/spec/MOFM2T
- **MOF Model to Text Transformation Language, v1.0**
 - OMG
 - www.omg.org/spec/MOFM2T/1.0/PDF

UML i SysML

- **Język inżynierii systemów SysML. Architektura i zastosowania. Profile UML 2.x w praktyce**
– S. Wrycza, B. Marcinkowski; wyd. Helion 2013
- **UML 2.0. Almanach**
– D. Pilone, N. Pitman; wyd. Helion 2007
- **Język UML 2.0 w modelowaniu systemów informatycznych**
– S. Wrycza, B. Marcinkowski, K. Wyrzykowski; wyd. Helion 2006
- **UML w kropelce, wersja 2.0**
– M. Fowler, wyd. LTP 2005
- **The Unified Modeling Language User Guide**
– G. Booch, J. Rumbaugh, I. Jacobson; wyd. Addison-Wesley Professional 2005
- **UML przewodnik użytkownika**
– G. Booch, J. Rumbaugh, I. Jacobson; wyd. WNT 2002

UML i SysML

- **Diagramy UML**
 - M. Wolski
 - wolski.pro/diagramy-uml
- **Notationsübersicht UML 2.5**
 - oose Innovative Informatik;
 - [www.oose.de/wp-content/uploads/2012/05/UML-Notations%
c3%bcbersicht-2.5.pdf](http://www.oose.de/wp-content/uploads/2012/05/UML-Notations%c3%bcbersicht-2.5.pdf)
- **UML**
 - Visual Paradigm
 - guides.visual-paradigm.com/category/uml
- **UML Modeling**
 - Visual Paradigm
 - www.visual-paradigm.com/VPGallery/diagrams
- **UML – Some metamodels**
 - [kompilatory.iiar.pwr.edu.pl/wiki/index.php/UML/Some_
metamodels](http://kompilatory.iiar.pwr.edu.pl/wiki/index.php/UML/Some_metamodels)
- **SysML Diagram Tutorial**
 - SysML.org
 - sysml.org/tutorials/sysml-diagram-tutorial

Visual Paradigm

- **Visual Paradigm Quick Start**
 - wyd. Visual Paradigm 2017
- **Visual Paradigm Tutorials**
 - Visual Paradigm
 - www.visual-paradigm.com/tutorials
- **Examples**
 - Visual Paradigm Community Circle
 - circle.visual-paradigm.com/diagram-examples

ATL

- **ATL**
 - Eclipse Foundation
 - eclipse.dev/atl
- **ATL/User Guide**
 - Eclipse Foundation
 - wiki.eclipse.org/ATL/User_Guide_-_Introduction
- **MMT/ATL Transformation Language (ATL)**
 - Eclipse Foundation
 - [wiki.eclipse.org/MMT/ATL_Transformation_Language_\(ATL\)](http://wiki.eclipse.org/MMT/ATL_Transformation_Language_(ATL))
- **Object Constraint Language**
 - OMG
 - www.omg.org/spec/OCL/2.4/PDF
- **OCL Operations Reference**
 - Eclipse Foundation
 - wiki.eclipse.org/Accelio/OCL_Operations_Reference

ATL

- **Basic Examples and Patterns**
 - Eclipse Foundation
 - eclipse.dev/atl/documentation/basicExamples_Patterns
- **ATL Transformations**
 - Eclipse Foundation
 - eclipse.dev/atl/atlTransformations
- **ATL Transformation Description Template**
 - ATLAS group LINA & INRIA Nantes, 2005
 - [eclipse.dev/atl/documentation/old/ATL_Transformation_Template\[v00.01\].pdf](http://eclipse.dev/atl/documentation/old/ATL_Transformation_Template[v00.01].pdf)
- **Specification of the ATL Virtual Machine**
 - ATLAS group LINA & INRIA Nantes, 2005
 - [eclipse.dev/atl/documentation/old/ATL_VMSpecification\[v00.01\].pdf](http://eclipse.dev/atl/documentation/old/ATL_VMSpecification[v00.01].pdf)
- **Transforming Models with ATL**
 - F. Jouault, I. Kurtev; LNCS 3844, 2005
 - www.researchgate.net/publication/213882827_Transforming_models_with_ATL

Acceleo

- **Acceleo**
 - Eclipse Foundation
 - wiki.eclipse.org/Acceleo
- **Acceleo - Getting Started**
 - Eclipse Foundation
 - wiki.eclipse.org/Acceleo/Getting_Started
- **Acceleo Features**
 - Eclipse Foundation
 - www.eclipse.org/acceleo/features.html
- **Acceleo Query Language**
 - Eclipse Foundation
 - www.eclipse.org/acceleo/documentation
- **Acceleo Operations Reference**
 - Eclipse Foundation
 - wiki.eclipse.org/Acceleo/Acceleo_Operations_Reference

Acceleo

- **Object Constraint Language**
 - OMG
 - www.omg.org/spec/OCL/2.4/PDF
- **OCL Operations Reference**
 - Eclipse Foundation
 - wiki.eclipse.org/Acceleo/OCL_Operations_Reference
- **Acceleo User Guide**
 - Eclipse Foundation
 - wiki.eclipse.org/Acceleo/User_Guide
- **Acceleo Best Practices**
 - Obeo
 - www.obeosoft.com/en/acceleo-best-practices

Temat następnej prezentacji

Projektowanie systemu informatycznego metodą MDA