Meta-Model Transformation using OCL and Patterns

Steve Schwartz
Senior Project Manager
Compuware Corporation
Agenda

- MDA promises
- Languages, transformations and refinement
- Raising the level of abstraction
- Types of transformations that do not constitute MDA
- MDA requirements summary
MDA Promises
MDA Qualities

• Portability
• Cross-platform Interoperability
• Platform Independence
• Domain Specificity
• Productivity
MDA Benefits

- Reduced cost
- Reduced development time
- Improved application quality
- Increased return on IT investments
- Rapid inclusion of emerging technologies
Classic Modeling and Development

- Designers & Developers
- Domain Knowledge
- Domain X Technology
- Platform Knowledge
- Classic Tools
- Applications
- Users
MDA Essentials

- Embrace and appreciate existing IT technologies
- Separation between, and reusability of, domain and platform expertise
- Quick adaptability of domain and technology changes
- Generation of working high-quality applications and integrations
Languages, Transformations and Refinement
Formal Languages

Lang. X expressed in Spec. A

MOF expressed in CWM

PPT expressed in MDA Promises
Transformations

Lang. X  
Spec. A
expressed in
 Transformation
Definition

Transformation

Lang. Y  
Spec. B
expressed in
 defined by
Example

UML

UML Classes to EJB trans. def.

Java

expressed in

defined by

expressed in

CRM model

Transformation

CRM EJBs
Transformation = Matching and Deriving Patterns

Lang. X

expressed in

matched patterns

Transformation Definition

defined by

transformation

Lang. Y

expressed in

derived patterns
Refinement Preserves Meaning and Derives Complex Patterns

Lang. X

expressed in

defined by

refinement

Lang. Y

expressed in

lower abstraction level

higher abstraction level
Transformations Conclusions

- Transformation = matching and deriving patterns
- The transformed languages can be equal or different
- Transformations can occur between different specifications or within one specification
- A refinement is a complete transformation that preserves meaning and derives more complex patterns than it matches
Raising the Level of Abstraction
“The entire history of software engineering is that of the rise in levels of abstraction.”

Grady Booch
The Limits of Software
September 2002
Hiding Complexity

- Abstraction eases the specification effort by hiding complexity.
Derivation of Intelligent Patterns

- Reuse of solutions
- Repeatable process
Coding Language Evolution

- Abstraction level has risen from 1GL to 4GL and back again to 3GL
- A 4GL is productive but lacks adaptability to realize specific requirements
- A current 3GL is mainly a set of standard libraries and frameworks, the coding syntax is less important
Modeling on Different Abstraction Levels

- In practice UML models are in different levels of abstraction
  - from analysis via design to implementation
- Translation vs. Elaboration?
- Automated or Creative process?
Productivity & Control

- Black-box automated refinement and hiding the detailed specification increases productivity but decreases fine-grained control
- Creative refinement and exposing the detailed specification increases fine-grained control but decreases productivity
- Law of preservation of misery?
Iterative Refinement

Abstract Specification

- Maintainable Refinement Definition
- Tuneable Detailed Specification

Refinement Definition

Detailed Specification

refinement
MDA Goal

Domain Experts

Platform Experts

Technology Solutions

Technology Selection and Tuning

MDA Tools

Applications

Application Developers

Users
MDA’s PIM, PSM and Iterative Refinement

Domain Model

Technology Patterns

Application

PIM

refine-able to less platforms

refine-able to more platforms

refinement
Different Abstraction Levels and Multiple PSMs

- Business Modelling language
- Technology patterns
- Application Modelling Languages
- Coding patterns

Domain Model

Application models

Application
Raising the Level of Abstraction is MDA

- Platform independent models
- Increased productivity because of automatically generated complexity
- Quality improvement because of enforcing intelligent patterns
- Separation between and reuse of domain and technology expertise
- Without loosing fine grained control
MDA
Requirements Summary
MDA Requirements

• High level of abstraction of PIM
  • No platform specific details
  • Refineable to many technologies

• Maintainable technology details
  • Changeable and layered PSMs
  • Maintainable technology and coding patterns

• Intelligent automated PIM to PSM refinement
  • High quality of technology patterns
  • Complete executable results
  • Incremental refinement