Implementing MDA Transformations in ATOM3

Wei He 2004.4.
Whe6@cs.mcgill.ca
Agenda

- Model Driven Architecture Review
- Project introduction
- Implementation details
- Demonstration
MDA Review

MDA software development life cycle

requirements

analysis

Mostly text

PIM

design

PSM

implementation

code

testing

code

deployment
MDA Review

- Building blocks in MDA
  1. Models
     1) Platform Independent Model (PIM)
     2) Platform Specific Model (PSM)
     3) Source Code
  2. Transformations
     1) PIM to PSM
     2) PSM to code
MDA Review

- Building blocks in MDA

  3. MDA specifications

  1) One or more standard, well-defined languages to write PIM

  2) One or more standard, well-defined languages to write PSM

  3) A language to write the definition of transformations between models
MDA Review

- Building blocks in MDA
  4. Transformation tools
MDA Review

Step 1. Build PIM

Step 2. Use transformation tool to generate one or more PSM from PIM

Step 3. Use transformation tool to generate codes
Project introduction
Project Introduction

- **Background:** develop an web-based ordering system for a breakfast service shop

- **Aim:** demonstrate the automated transformations from PIM to PSM and from PSM to source code
Project Introduction

Implementation:

1. Map classes to tables by Graph Grammar transformation in ATOM3
2. Transform tables to SQL
3. Create tables in mySql DBMS

Environment: ATOM3, mySql
Implementation Details
Implementation Details

- PIM Meta model—Class Diagram
Implementation Details

- PSM Meta model—Table Diagram

![Table Diagram](image)
Implementation Details

- PIM model

```
BreakfastOrder
  deliveryTime
  Date
  ;
  deliveryAddress
  String

Customer
  name
  String
  ;
  address
  String

Breakfast
  number
  Integer
  ;

StandardBreakfast
  name
  String
  ;
  price
  Real

MyChange
  quantity
  Integer
  ;

Comestible
  name
  String
  ;
  price
  Real

Part
  quantity
  Integer
  ;
```
Implementation Details

PIM to PSM transformation – GG rules

Rule 1. Handling association classes (LHS)
Implementation Details

PIM to PSM transformation – GG rules

Rule 1. Handling association classes (RHS)
Implementation Details

- After applying rule1:
Implementation Details

- PIM to PSM transformation – GG rules
  
  Rule 2. Handling associations between classes (LHS)
Implementation Details

- PIM to PSM transformation – GG rules
  
  Rule 2. Handling associations between classes (RHS)
Implementation Details

- After applying rule2:
Implementation Details

- PIM to PSM transformation – GG rules
  
  Rule 3. Handling inheritance between classes (LHS)
Implementation Details

- PIM to PSM transformation – GG rules
  - Rule3. Handling inheritance between classes (RHS)
Implementation Details

- After applying rule3:
Implementation Details

- PIM to PSM transformation – GG rules

Rule 4. Mapping classes to tables (LHS)
Implementation Details

- PIM to PSM transformation – GG rules

  Rule 4. Mapping classes to tables (RHS)
Implementation Details

- After applying rule4:
Implementation Details

- PSM to Source code transformation
  1. Iterate over table blocks
  2. Generate “Drop Table…” and “Create Table…” for each table block
  3. Output CreateTables.sql
Implementation Details

- Create tables in mySql DBMS
  feed CreateTables.sql into mySQL
Demonstration
This project is based on the example given in

<<MDA Explained
the Model Driven Architecture:
Practice and Promise>>

by Jos Warmer, Anneke Kleppe and Wim Bast