MDA and OptimalJ

Martin Novák
<mn@puakma.net>
What Is Presented?

- Model Driven Architecture (MDA)
- Tools supporting MDA development process - Compuware OptimalJ
• Standard issued by OMG
• Based on previous OMG work like UML
• Complete description of development process
OMG Standard

- Build on solid foundation of existing standards
- UML
- XMI - XML Metadata Interchange
- MOF - Meta Object Facility
What MDA addresses?

- Portability, reusability on several development platforms - EJB, SOAP, CORBA, ...
- Exchange of data based on standard communication protocols
- Again - tool for the topmost enterprise development
What MDA Addresses?

• It looks like MDA tends to use only the most complex and the most complicated tools on the market

• Also looks like MDA will be very overcomplicated, and very expensive
What MDA Addresses?

• But also MDA is trying to simplify steps normally done more times
• Makes UML finally usable
MDA Bases

- Model is build in three layers
- Platform Independent Model (PIM)
- Platform Dependent Model (PDM)
- Code Model
Models

- Models are transformed from PIM to Code using transformations
- Due to transformations we can make from domain model both CORBA, and SOAP
Platform Independent Model - PIM

- Domain analysis
- Using UML
Platform Dependent Model - PDM

• Used for modifying platform dependent model
• You will change different properties, and make different functions in CORBA, and SOAP
Again using transformations you can create skeleton/or usable web application or SOAP module.
OptimalJ

• Tool for MDA support
• Professional edition is integrated with Netbeans Platform, and developer edition with IntelliJ IDEA
• Doesn’t run on MacOS X although written in Java
• Bad experience with company support
Workflow

- Domain Model
- Application Model
- Code Model
MOF - Meta Modeling Facility

- Developed by OMG
- OptimalJ uses v 1.4
- used for: generate repository API for each model
- importing/exporting models
- enables transformations between models
Domain Model

- Editing abstract domain model
- Based on class diagrams
- There is also some model checking
Domain Model
You can import Model from:

- UML
- JDBC
- EJB
- CORBA (IDL)
- WSDL schema
Application Model

- Lower level model for editing of model of specific technology
- EJBs
- Relational database model
- SOAP
- JSP/servlet webs
- CORBA
Database Model

- Relational database editor
- Refining tables, types, etc...
- Further transforms to SQL
- Netbeans has some SQL tools
DAO Model

- Database Access Objects
- Depends on the vendor - because uses class code injection
EJB Model

• Supports EJB 1.1 and 2.0
Web Application

- Generates full JSP/servlet web sites using EJBs
- Can generate also WAP webs
Web Services

- SOAP
- Can either generate or consume web services
OptimalJ Architecture Edition

• Enables you to develop patterns, and transformations used in OptimalJ

• You can also modify code generated by OptimalJ
**Code Model**

- Is the generated code from application model
- Java and web sources can be further edited in Netbeans IDE or IDEA IDE
OptimalJ - Cons

• Doesn’t allow to create any simple Java model from domain - unusable for lot of cases

• Didn’t work properly without deep knowledge - even using the simplest tutorial made some errors, and it was very unclear what does it mean

• J2EE only tool
OptimalJ - Pros

- Quite nice domain model editor
- Nicely transfers between domain and database model
Conclusion

• Promising technology to the future
• BUT - there should be something more than OptimalJ offers today