



IBM Software Group

Model Driven Development

A horizontal bar with a series of colored squares (green, yellow, red, purple, cyan) followed by a grayscale image strip containing icons of a building, a refresh symbol, a person's face, and a hand.

Rational. software

© 2007,2009 IBM Corporation

DISCLAIMER

This talk is presented in the view of the author(s) rather than IBM

Why Model Driven Development (MDD) ?

Model-driven development is a style of software development where the primary software artifacts are models from which code and other artifacts are generated.

A *model* is a description of a system from a particular perspective, omitting irrelevant detail so that the characteristics of interest are seen more clearly. A Model must be machine readable so that it could be used to generate other artifacts (documentation, code , etc)

How MDD works

Models are used mostly as *sketches* that informally convey some aspect of a system or *blueprints* that describe a detailed design that you manually implement.

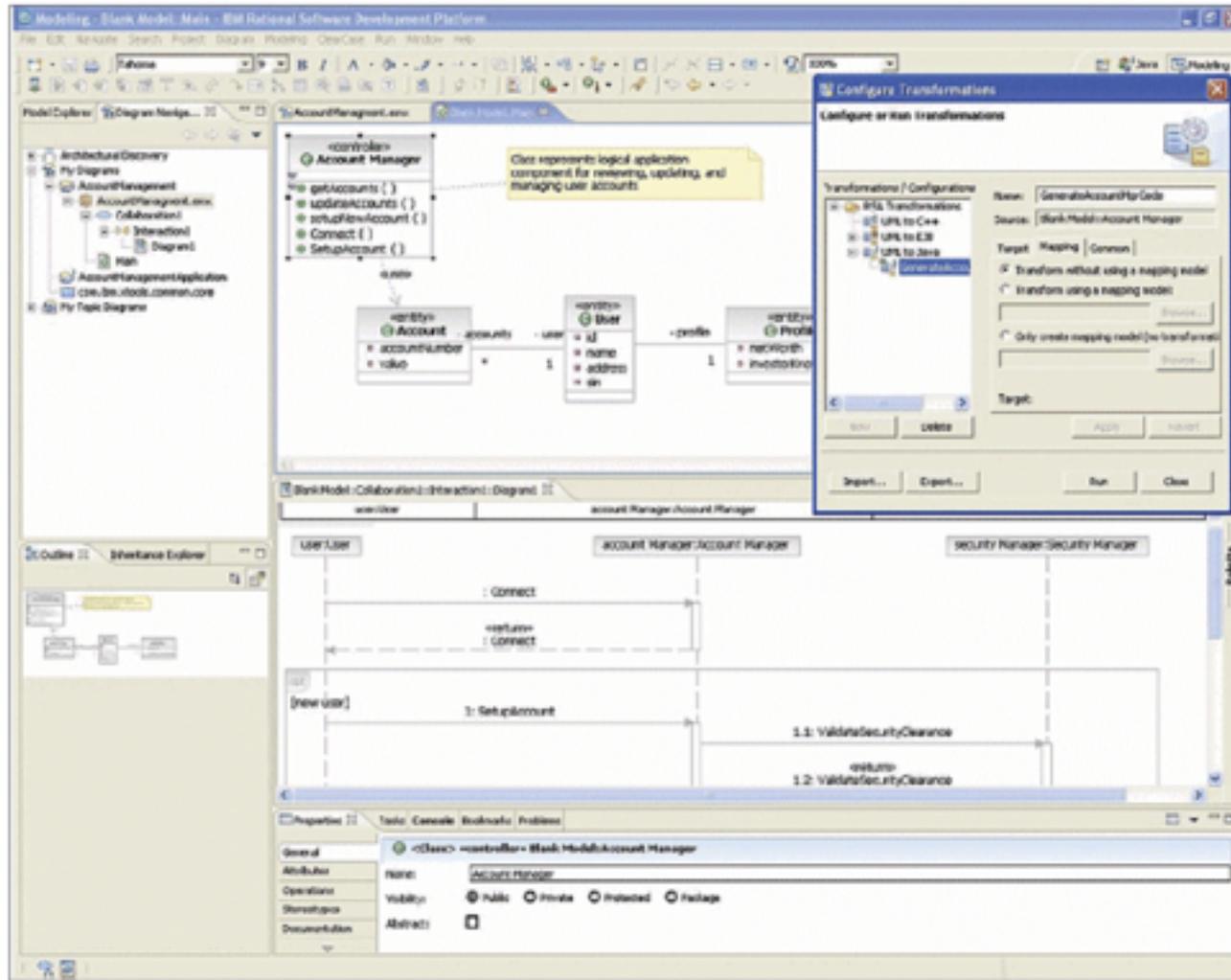
Using models as documentation and specification is valuable, but it requires strict discipline to ensure that models are kept up to date as implementation progresses.

The benefits of MDD

- Increased productivity
- Maintainability
- Reuse of legacy
- Consistency
- Reusability
- Improved stakeholder/design communication

- Rational Software Architect
 - ▶ Integrated development environment specifically designed for modeling
- RSA has the following features that particular relevant to MDD
 - ▶ UML 2.0 editor with refactoring support
 - ▶ Support for UML 2.0 profiles
 - ▶ Pattern infrastructure with pattern library
 - ▶ **Transformation infrastructure** with sample transformations

RSA Overview

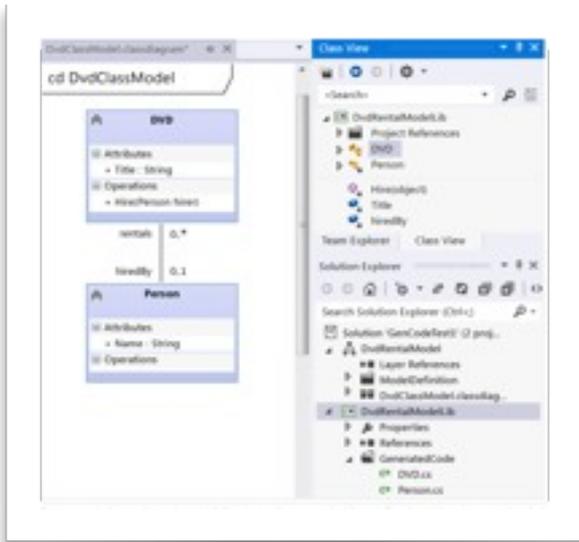
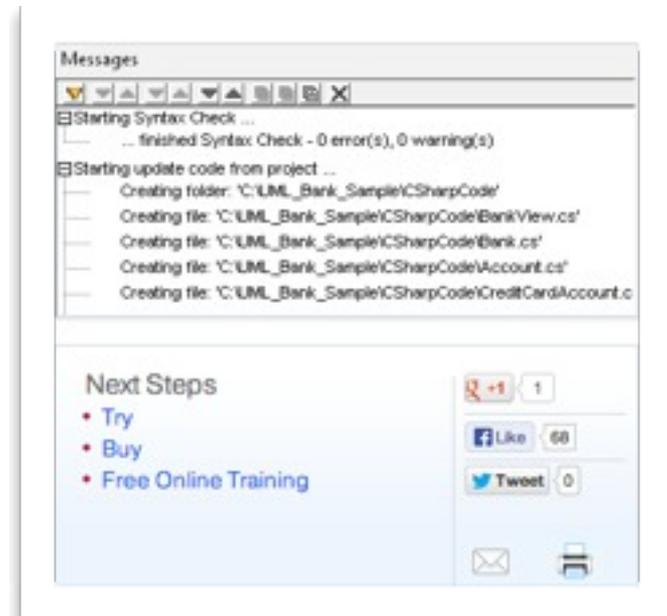
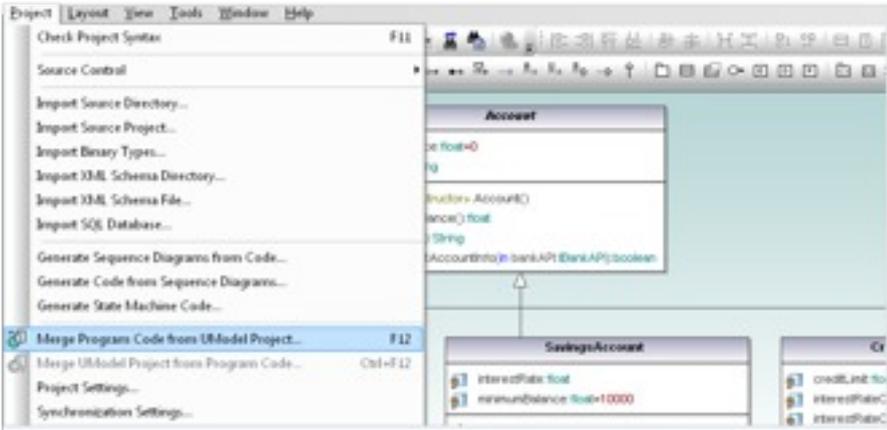


UML 2.0 editor

Other options

Altova UModel 2013

<http://www.altova.com/umodel/uml-code-generation.html>



Microsoft Visual Studio 2012

<http://msdn.microsoft.com/en-us/library/vstudio/ee329480.aspx>



UML Profiles

UML profiles allow you to customize the language for a particular domain or method.

UML profiles introduce a set of stereotypes that extend existing elements of UML for use in a particular context.

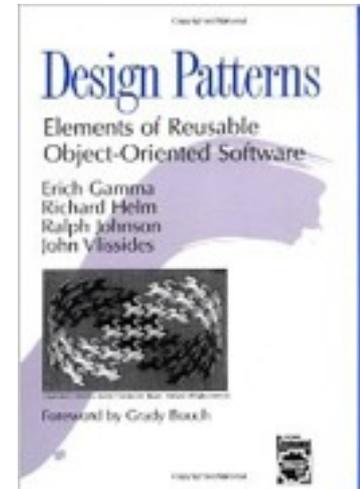


Example Application of the RUP Analysis Profile

Design Patterns ?

Design Pattern:

A design pattern is a general **repeatable** solution to a **commonly-occurring problem** in software design.

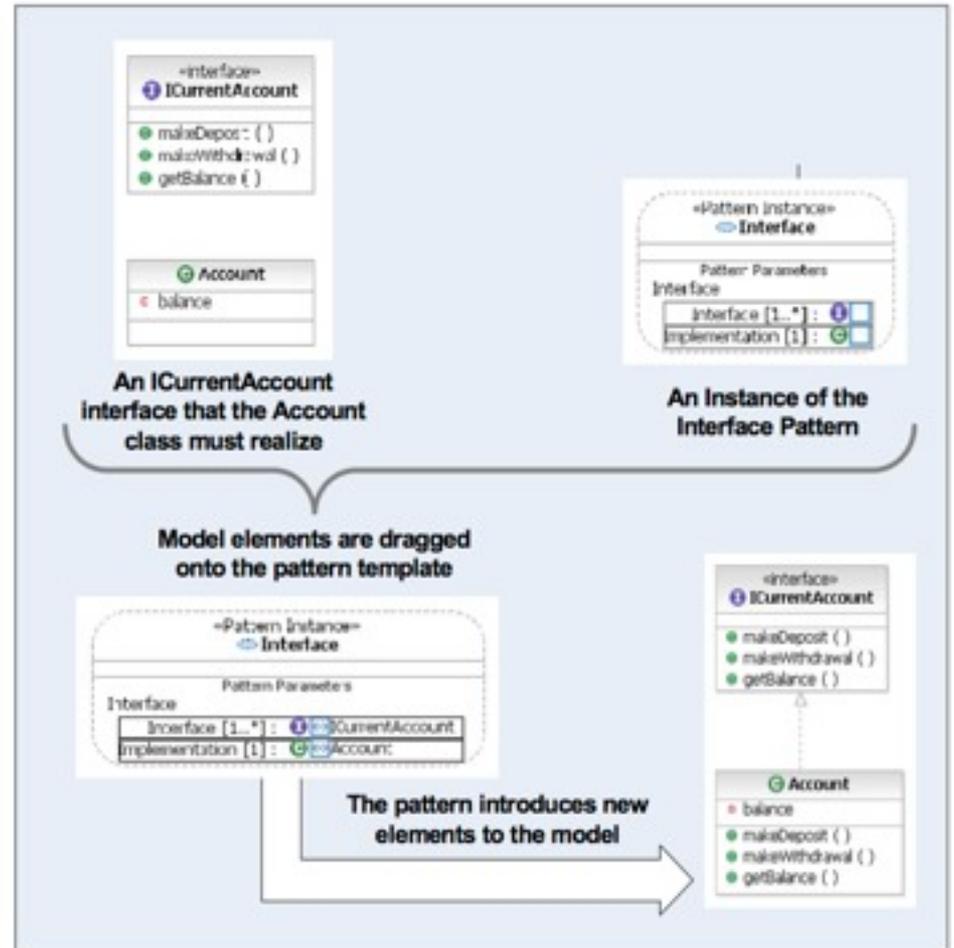


RSA Patterns

Patterns are solutions that solve a recurring problem in a particular context.

RSA ships with a set of patterns including the **Gang-of-Four** patterns.

RSA also provides a patterns infrastructure so that developers can build their own patterns.

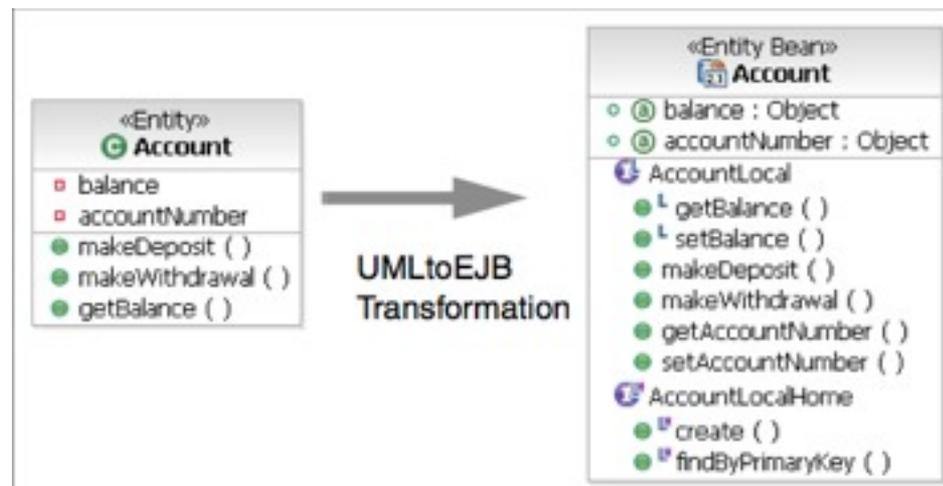


RSA Transformation

Transformations support a layered, model-driven approach to development.

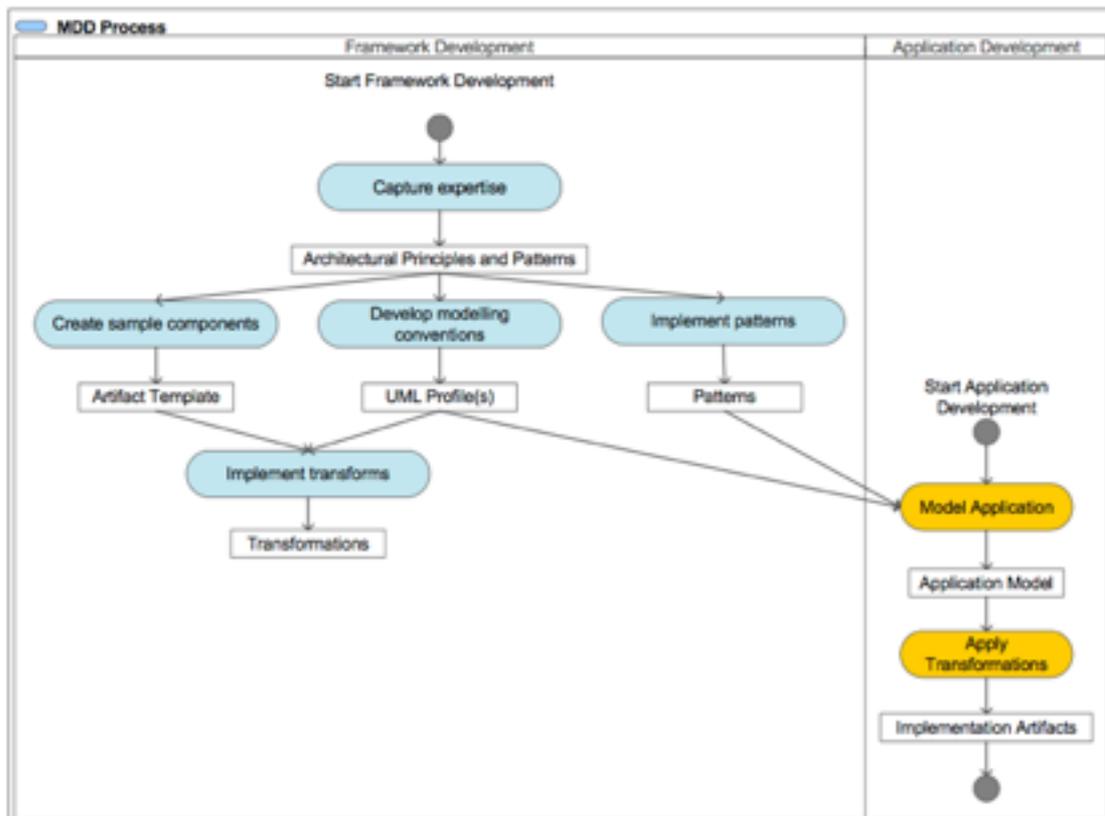
They automate the transition between models at different levels of abstraction (for example, from analysis to design) and ultimately to code.

RSA ships with a sample set of transformations, including a UML to EJBTM transformation. RSA also provides a **transformation infrastructure** so that developers can build their own transformations and extend the provided transformations.



The MDD Process

- ▶ **Expertise Capture and Automation:** This is where we build the MDD framework that partially automates the development of software that follows a particular architectural style.
- ▶ **Application Development:** This is where we apply our chosen MDD framework to build software components, applications, and solutions.



Proxy Pattern

Provide a surrogate or placeholder for another object to control access to it.

Challenge

- Authentication process is quite slow. Is there any way to improve its performance ?

- Can we make the enhancement transparent to existing clients ?



Problem

- You want to add a middle-layer between clients and your system.
 - ▶ Access control
 - ▶ Performance enhancement
- The implementation must be transparent to existing user



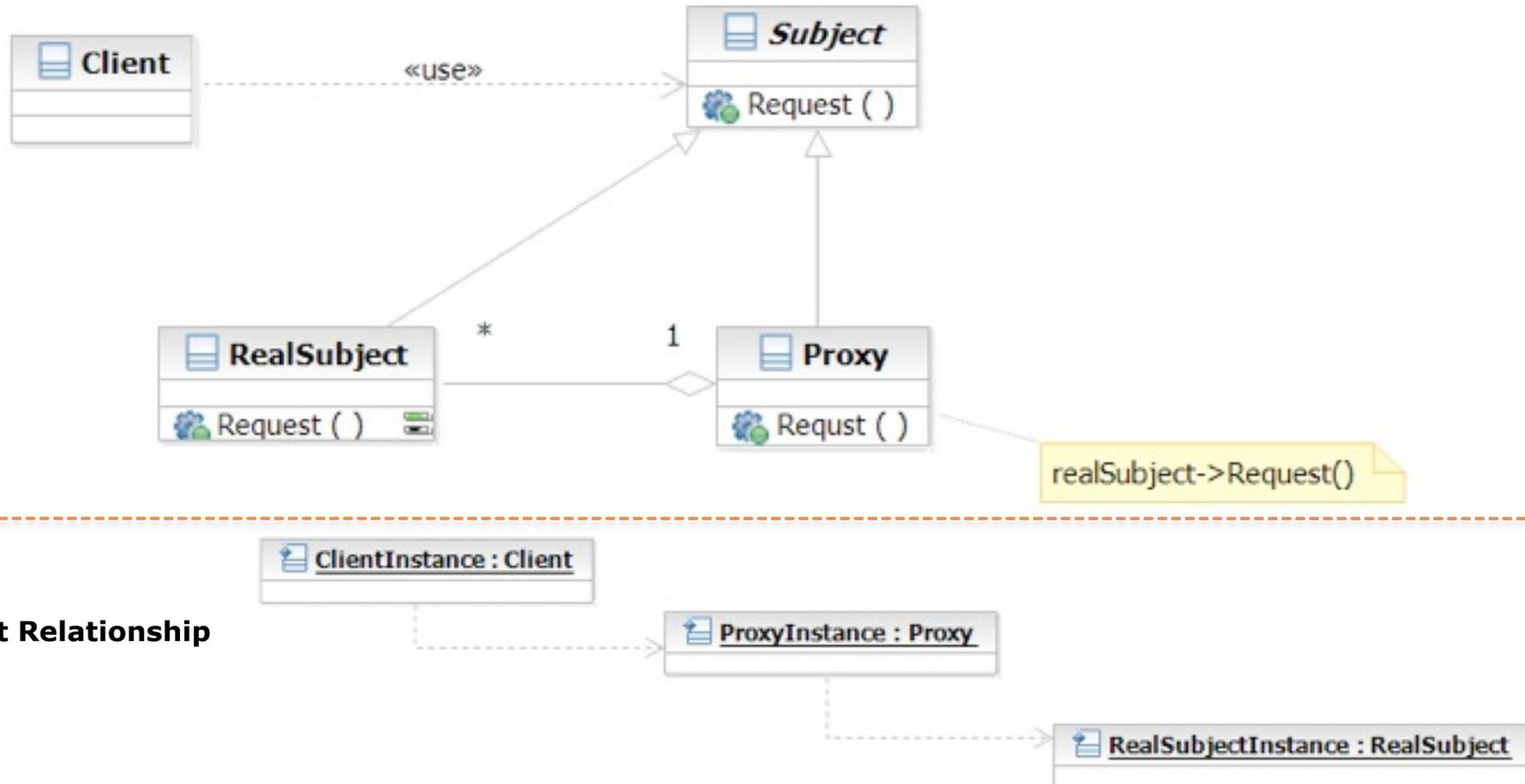
Think ...

- How to implement the access control?
- Can you do something before client program accesses your resource

- Target:
 - A proxy that can be act as a gate-keeper of existing resource



Structure/Participants



Applicability

- Uses of Proxy pattern
 - Remote proxy
 - Reversed proxy
 - Protection proxy

Consequence

- Indirect access of resources
 - You can always monitor/filter the access request
- Resource control optimization



RSA Demo

Scenario :

A security gateway (proxy) that check whether the input username/password contains dictionary attack

- > if success, pass the original login request to the target authentication/authorization server
- > if failed, reject the transaction immediately