Practical Model-Driven Development with the IBM® Software Development Platform

Osmond Ng (ong@hk1.ibm.com)
Technical Consultant, IBM HK SWG
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
The Challenges of Enterprise Software Development

- Ever-increasing complexity in the operational environments
  - J2SE, J2EE, .Net
  - Web, Handhelds, disconnected
  - Legacy integration, modernizing

- Ever-expanding choices to make on development solutions
  - Programming Languages, scripting Languages
  - IDE’s, testing tools

- Ever-changing nature how software gets created
  - Globally development teams
  - Outsourcing
  - Compliance and Regulations

More Layers, More Servers, More Frameworks, More “Moving Parts”,
More Complexity
Overview

- The Challenges of Enterprise Software Development
- **Modeling: The Key to Managing Software Complexity**
  - What is Model-Driven Development?
  - What is Model-Driven Architecture?
  - What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Modeling: The Key to Managing Software Complexity

- Modeling is the standard approach in engineering to
  - Manage Complexity
  - Mitigate Risk
- Software development is the same as every other kind of engineering in this respect

Maybe you have to

But then, maybe you should

Well, maybe you shouldn’t'
Modeling: The Key to Managing Software Complexity

- What people are saying today about software modeling*
  - “…ensures that customers get what they ask for”
  - “…allows me to create domain-specific areas of expertise and bring in the right people at the right time”
  - “…allows me to better understand my enterprise at different levels of detail for different stakeholders”
  - “…higher levels of reuse, reduce overall costs

Those who have adopted software modeling are finding that it improves technical quality, reduces business costs, and better manages risk.

Source: IBM whitepaper, The Value of Modeling
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
What is Model-Driven Development (MDD)?

- The encapsulation of business logic and industry best practices into models
- The use of these models for application development, code generation, testing, and maintenance

- Modeling helps you work at higher levels of abstraction
- Higher levels of abstraction lead to higher productivity
Model-driven development is aided by a common *language* across all stakeholders

- Unified Modeling Language (UML) is the standard language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
- UML allows software architects, designers and developers to specify, visualize, construct, and document all aspects of a software system
UML – The Language of Model-Driven Development

- Key leaders from IBM Rational led the original development of UML
- Today, UML is managed by the Object Management Group (OMG)
UML – The Language of Model-Driven Development

The Evolution of UML

UML 1.1 (OMG Standard)
- Rumbaugh
- Booch
- Jacobson
- Foundations of OO (Nygaard, Goldberg, Meyer, Stroustrup, Harel, Wirfs-Brock, Reenskaug,...)

UML 1.3 (extensibility)
- 1998

UML 1.4 (action semantics)
- 2001

UML 1.5
- 2003

UML 2.0 (MDA)
- 2003

IBM Software Group
What is Model-Driven Development?

Different Stakeholders, Different Models

- Business Model
  - Visualization of business processes
- Use Case Model
  - Visualization of functional requirements
- Analysis Model
  - “What” the system must do to realize the functional requirements
- User Experience Model
  - Visualization of user interaction with the system
- Design Model
  - “How” the system will realize the functional requirements
- Data Model
  - Visualization of persistent storage
- Implementation Model
  - Visualization of the code
What is Model-Driven Development?

Model-driven development is not “one size fits all”

- You can model at various levels of abstraction and detail
- The key is to choose the level that’s right for your project and team
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
What is “Model-Driven Architecture”?

- An OMG initiative
- Different levels of abstraction via multi-models
- Platform Independent
- Platform Specific
- Transform from one level to another

**Platform-Independent Model**

**Platform-Specific Model**

- Business functionality and behavior
- Standard mapping

**Design of the solution based on a specific platform like J2EE**

**Platform Architect**
What Is “Model-driven Architecture”?

- MDA is a particular style of model-driven development
- MDA defines a philosophy for how models should be used in the development process
  - Distinct models at particular levels of abstraction
  - Notion of model transformations (model-to-model, model-to-code)
  - Strong separation of concerns across spectrum from business to implementation
- MDA is not yet itself an OMG standard
  - It’s based on UML and other OMG standards
  - There is a set of specifications for MDA
- MDA is in an early adopter phase – customer stories are emerging
- Initial MDA guide over-simplified modeling abstractions (CIM, PIM, PSM)
- Of all the ways to practice model-driven development, MDA is at the top of the food chain—the most sophisticated, and potentially the most powerful
- Rational Software Architect supports a wide and granular set of MDD styles of development, with MDA being but one
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?*
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
What do we mean by “Practical Model-Driven Development”

- Why don’t more developers model their software?
  - Is it cultural?
    - “Just give me an editor and a debugger”
    - “We can’t afford modeling tools”
    - “Modeling is documentation — I hate doing documentation”
    - “In the past we’ve tried and failed.

- The key reasons are the perceptions that
  - Modeling is **hard**
  - Modeling will **slow me down!**
### Practical Model-Driven Development

<table>
<thead>
<tr>
<th>Activity Diagrams</th>
<th>Business Process models</th>
<th>Asset Based Development</th>
<th>OMG MDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Executeable Models</td>
<td></td>
</tr>
<tr>
<td>UML code Visualization and edit</td>
<td>UML EJB transforms</td>
<td>Code Analysis</td>
<td></td>
</tr>
<tr>
<td>Code Wizards, Class explorer</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **To a developer **practical model-driven development**
  - Requires low up-front investment
  - Results in high short-term benefits
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?

**Practical Model-Driven Development with Rational Software Architect**
- Browse and Topic Diagram Creation
- Static Sequence Diagram Creation
- Architectural Discovery Diagrams
- Structural Analysis Code Review
- J2EE Development Using UML Transformations
A Quick Look at the New Rational Design and Construction Tool Set

- UML 2.0 modeling
- Pattern/Transform Authoring
- Reusable Asset Browser

- Web Development
- Web Services Development
- Rich Client Development
- XML & Database Tools
- Java Generation Tools
- Unit Test

- UML Language Transforms
- Structural Review & Control
- C/C++ Development Tools

- J2EE/EJB & Portal Development
- Component Testing
- Code Review & Runtime Analysis
- UML Visual Editors
- Configuration Management
Getting to Know Rational Software Architect

- Demo: Overview of Rational Software Architect
A Quick Intro to UML for Java developers

A Class is represented by a rectangle with compartments

Stereotype – what kind of a UML class is it?

Name Compartment

```java
public class SubClass {
    private int priority;
}
```

Attribute Compartment

```java
public class SubClass {
    private int priority;
}
```

Operation Compartment

```java
public class SubClass {
    private int priority;

    public String echo(String aString) {
    
    public String identifyMe() {
    
    public String workWith{
```
A Quick Introduction to UML for Java developers

Generalization (inherits)

Realization (implements)

Association
A Quick Intro to UML for Java developers

Use

```java
public class SubClass extends BaseClass {
    public String workWith(UsedClass usedClass) {
        CoreClass coreClass = new CoreClass();
        return "";
    }
}
```
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Overview

- The Challenges of Enterprise Software Development
- Modeling: The Key to Managing Software Complexity
- What is Model-Driven Development?
- What is Model-Driven Architecture?
- What do we mean by “Practical Model-Driven Development”?
- Practical Model-Driven Development with Rational Software Architect
  - Browse and Topic Diagram Creation
  - Static Sequence Diagram Creation
  - Architectural Discovery Diagrams
  - Structural Analysis Code Review
  - J2EE Development Using UML Transformations
Summary

- Modeling is the key to managing the complexity and risks associated with today’s enterprise software systems.
- Model-driven development helps teams work at higher levels of abstraction, which leads to higher productivity.
- “Practical model-driven development” refers to tools and techniques that allow developers to rapidly gain the benefits of model-driven development with little up-front investment.
- Rational Software Architect is IBM’s premier design & development tool enabling model-driven development and model-driven architecture. It incorporates all the capabilities in Rational Application Developer for WebSphere for building scaleable Web, Web services, Java, J2EE and portal applications.
Resources

- Learn about the technology
  - IBM Rational® Software Architect Product Page
      - Contains pointers to training, tutorials, articles, and many other resources related to model-driven development and model-driven architecture

- Hands-on learning
  - IBM Rational Software Architect Trial Version
  - Other IBM Rational Trial Versions are at http://www.ibm.com/rational
    - Click on “Trials and Betas”

- Modeling and MDA Books
  - Ahmed, K. Developing Enterprise Java Applications with J2EE and UML
  - Eeles, P. Building J2EE Applications with the Rational Unified Process
Questions
Thank You