Introduction to Model-Driven Development (MDD)

The Virtual Enterprise (VE)

September 18, 2008
Agenda

- Introduction
- Evolution of Abstraction
- Model Driven Development
- The Virtual Enterprise
- Demo
- Conclusions
- Q & A
Introduction

• Speaker
  – Ashok Nare, CTO, Intelliun Corporation

• Company Overview
  – Intelliun, founded in 1999, is a leading innovator in the area of Model Driven Development
  – Intelliun’s Model Driven Development platform, *The Virtual Enterprise*, facilitates the development of web-based agile business solutions by capturing the business logic in technology independent, UML based visual models
  – Intelliun Services Group provides a comprehensive set of services for the development of custom web-based applications using Intelliun’s MDD platform.
**Evolution of Abstraction**

- Abstraction is concentration on relevant aspects of the problem and ignoring those that are not important.
- **Evolution of programming languages**
  - Machine language to Assembly language to higher level languages such as C++, Java, C#, Scripting, etc.
  - More time was spent on understanding “how” to solve the problem in early languages (understand the language).
  - Each language raised the level of abstraction by hiding low level details.
- **Evolution of tools, frameworks and application servers**
  - Abstraction and reuse of common services.
- **Focus on solution to the problem by working with concepts and terms that are familiar to the problem space and ignoring the low level details.**
- **Abstraction is the key to building complex software.**
Model Driven Development (MDD)

• What is MDD?
  – A software development approach that uses models to capture application logic during the development of end-to-end enterprise applications
  – Forrester’s Definition:
    “An iterative approach to software development where models are the source of program execution with or without code generation.”

• MDD Objectives
  – Raise the level of abstraction for application development
  – Reduce development time
  – Improve application quality while reducing testing time
  – Reduce maintenance cost and Total Cost of Ownership of enterprise applications
Model Driven Development (MDD)

• How
  – Use models to implement application logic
  – The domain model (in Abstract Design) is the implementation model
  – Use automation to generate executables from the implementation model in runtime or build time

• Approaches
  – OMG Model-Driven Architecture (MDA)
  – Executable Models
Software Development

1. Functional Requirement
2. Requirement Analysis
3. Abstract Design
4. Architecture
5. Technologies & Tools
6. Detailed Design
7. Coding
8. Executable
9. Non-functional Requirement
Flavors of MDD
OMG Model Driven Architecture (MDA)

Pros
- Reusable PIM models
- Methodical approach to software development
- More direct input from Application Analysts/Architects
- Marginal reduction in development time

Cons
- PIM models cannot be validated
- PSM are platform/architecture dependent, which makes up for the majority of the artifacts
- Transformation/code-generation is an integral part of the development process
- Generated code still require further development
Flavors of MDD
Executable Models

- Functional Requirement
- Requirement Analysis
- Abstract Design
- Technologies & Tools
- Architecture
- Detailed Design
- Coding
- Executable
- Non-functional Requirement
Flavors of MDD
Executable Models

Pros
- Abstract Design is 100% semantically complete and reusable
- Models can be immediately validated
- Significant reduction in development and maintenance time
- Code generation is optional and after-the-fact

Cons
- Supports a single architecture
- Depends on runtime environment
- No access to generated code
The Virtual Enterprise (VE) is a comprehensive platform for the rapid development of agile business solutions using Model-Driven Development (MDD). Built on top of J2EE, VE offers a portable execution stack for the delivery of highly scalable business applications and Web services.

The Model is the Executable
Architecture

Support for multiple devices including Web Browser, PDA, Web Services

All application logic is captured in UML notation (class and activity diagrams)

Support for wide range of relational databases and auto-generation of database schema & OR-Map
# Features

**VE/Designer**
- Develop web applications using UML models
- Instant execution of UML models and validation of application logic
- Dynamic generation of the web interface and Web services
- WYSIWYG web personalization
- Dynamic generation of object-relational database mapping
- Formula auto-completion
- Support embedding Java code and JAR files
- Support embedding hand-coded SQL statements/stored procedures
- Unit/remote testing framework

**VE/Server**
- Runs on any J2EE web and/or application server
- Runs on any Java supported platform including Unix, Linux, Windows and AS/400
- Supports wide range of relational databases including MS-SQL, Oracle, DB2/UDB, MySQL, Pervasive, and Sybase
- Supports SOAP and WSDL in both client and server scenarios
- Supports REST in both client and server scenarios
- Supports JMS for messaging and events
- Provides full localization
Applications

- VE has been used to build applications in the healthcare, finance, transportation, philanthropy, home-improvement, retail, telecommunication and engineering industries.

- Example Applications:
  - ComplianceSet: Sarbanes-Oxley Compliance ASP
  - Safeguard: Full Public Storage Management System
  - MicroEdge Portico: Grant Management
  - Attorneys Website, Intranet, Bios, and Proposal Management (deployed at some of the largest law firms in the world).
  - TotalChart: Surgeon Practice Management ASP

- Reusable Components:
  - Reporting
  - Discussion Board
  - Scheduling
  - User-customizable Portals
  - Document Management
Case Study – Self Storage Management System

- **Key Modules**
  - Corporate Administration
  - Regional Administration
  - Point of Sale
  - Product Catalog
  - Inventory Management
  - Pricing
  - Customer Management
  - Call Center Integration
  - Online Sales

- **Key Metrics**
  - # of Packages = 66
  - # of Objects = 609
  - # of Database Tables = 200
  - # of Integration Points = 6
Approach Summary

- Application logic is captured in platform independent UML models
- Models are immediately executable as they’re developed (no code generation, compilation, and deployment required)
- The development focus is always on the domain model, where interface and persistence is auto generated and can be later customized
- Code generation is optional and after-the-fact
- Code generation is done via templates that can be customized to control language, coding style, design patterns, and technology choices
Advantages of MDD

• Captures application logic in platform independent UML models
• Simplifies web development by reducing the number of required skills in the underlying technologies, specifications and standards
• Provides immediate validation of business requirements
• Improves communication among stakeholders
• Protects business IP investments from evolving technologies
• Radically reduces the development, time, cost and effort of business applications and Web Services
• Increases application agility to better align with continuously changing business needs
• Reduces QA time while improving application quality
Questions

Thank You

sales@intelliun.com
www.intelliun.com