A Model Driven Development Platform using Eclipse, EMF, UML and more

Sridhar Iyengar
IBM Distinguished Engineer
Chief Technical Strategist
IBM Rational Software
Outline

- Key Trends in Application Development
- Application Life Cycle Management
  - Integration of full lifecycle development, deployment and management tools
  - The Rise of UML, MOF, XMI standards and their pragmatic implementation at eclipse.org
  - The arrival of model visualization and code generation in open source frameworks (EMF, GEF and UML projects)
- Building an MDD platform using Eclipse
- A Look to the Future
Some Key Trends in Application Development - 1

- The rise of open source (eclipse) and open standards (W3C, OASIS and OMG) based tools

- From individual tools to a suite of tools – ‘End-end application lifecycle management’

- From multiple tools platform (one for each vendor) to a unified Eclipse open tools platform with support for rich data/metadata interchange

- The move to ‘Asset Based Development’
  - The use of patterns, recipes and code/model transformation frameworks to speed development
  - Arrival of RAS – Reusable Asset Specification for managing asset metadata

- Enterprise customers moving to integrated tool suites gradually – still many gaps to close
Some Key Trends in Application Development - 2

- The concept of end-end Application Life Cycle Management taking hold
- Evolution of Software Development as a Business Process
  - IBM, Borland and now Microsoft are showing increased focus on Software Development as a process’
  - ‘Software Development as a Business Process’ is now highlighted by IBM, Borland and others
- Vendors announce ‘new & improved’ platforms – with tools customized by role – yet integrated
  - Eclipse Open Source & Open standards based
    - Borland SDP - Software Delivery Platform (2005)
  - Microsoft Visual Studio Team System
- Tools for each user role
  - Business Analyst, IT Architect, Data Architect, Developer, Tester, Manager, Executive etc
An Enterprise Application Development Life Cycle

*Architecture Centric, Business driven, Complex Life Cycle, Many Tools*

- Discovery and Transformation
- Modeling Architecture, Object, Biz...
- Add Business Logic
- Build/Wrap Components
- Assemble & Test Components
- Configure & Deploy Components
- Manage Component
- Application Platforms (J2EE, SOA..)
- Acquired Components
- Models, Metadata, Mappings enabled Integration Platform

- Rapid
- Rigorous

And do this with quality in a distributed environment
Model Driven Development & Deployment

**Design/Build**
- Business Modeling (BPD, UML)
- IT Modeling (UML, SQL, XSD)
- J2EE/Web Services Development Wrapping Orchestration (J2EE)
- Deployment J2EE App Svr Web Services

**Run/Manage**
- Management Component Mgmt App Mgmt

**Specific metadata Models**
- BPM
- Biz Rules
- UML2
- SQL
- C++
- SPEM

**Traceability Links and Transformations**
- UML2
- J2EE
- BPEL
- WSDL
- XML
- UML2
- J2EE
- EMF
- SQL

**Serve up models, Components, processes On Demand**
- J2EE
- DCM
- CIM
- SI
- J2EE

**Technical Details**
- Specific metadata models
- Traceability links and transformations (profiles, metamodels, Code Gen Templates)
- Serve up models, Components, processes On Demand
The IBM Software Development Platform on Eclipse

- Analyst
  - WebSphere Business Integration Modeler & Monitor
  - Rational Software Modeler
- Architect
  - Rational Software Architect
- Developer
  - WSAD IE
  - Rational Application Developer
  - Rational Web Developer
- Tester
  - Rational Functional & Manual Tester
  - Rational Performance Tester (2Q05)
- Deployment Manager
  - Tivoli Configuration Manager
  - Tivoli Monitoring

Customer Extensions

Rational Team Unifying Platform

Project Manager

Rational Portfolio Manager

3rd Party ISV Tools
Borland Core Software Delivery Platform on Eclipse: February 2005 announcement

Announcing Borland Core SDP

The enabling technology for Borland Software Delivery Optimization (SDO), as part of the SDO vision, Borland Core SDP offers the first cost-effective and integrated process and roles centric platform for application lifecycle management. It is designed to enable IT organizations to reduce their software delivery risks by transforming their software development into an organized business process for the delivery of high quality software on time, on budget within scope, and at acceptable quality levels.

"Borland Core SDP takes a substantial step toward evolving the culture of software development from what is now largely ad-hoc individualism to a more cohesive, linked approach."

—Malcolm Weamer, research director of application development and deployment at IDC

DOWNLOAD THE BROCHURE | GET THE FAQ | READ THE PRESS RELEASE

Core Analyst | Core Architect | Core Developer | Core Tester

Core SDP Platform Services

Visibility | Foundation Services | Teamwork & Communication

PROCESS- AND ROLE-CENTRIC

Proc process solution with the ability to improve processes

SECURE AND DISTRIBUTED DEVELOPMENT

Borland Core SDP provides secure access to software
Microsoft Visual Studio Team System 2005

Have we solved the tools integration problem?

- The work began in earnest in 2001 with the formation of eclipse.org and IBM’s donation to the open source project.

- We are in the ‘second inning’: Eclipse Foundation members are uniquely positioned to solve the problem.

- Interestingly, we are getting encouragement and help from customers to increase the pace.
  - Example: Global Bank ‘Development Highway’ Case Study.
  - Their complaint ‘Why can’t you all get along and work together to solve our problem’!
Implementing Model Driven Development Lifecycle using Open Source and Open Standards

Case Study – ‘The Development Highway’ Project at Global Bank
Basic Principles used in the ‘Development Highway’

- **Move to Model Driven Development and Service Oriented Architecture**
  - Flexibility and separation of concerns between Business and IT domains
  - Consistent use of models and metadata across tools
  - Consistent use of development processes across globally distributed development teams
- **Use of commercial tools and application platforms that support open standards**
  - J2EE, Web Services (W3C, WS-I, OASIS) and MDA (OMG MOF, RAS, XMI) standards
  - Use Eclipse as the Tools Integration Platform
- **Reduce custom development of tools as much as possible and move to open standards**
- **Multiple vendors ‘strongly encouraged’ to ‘co-operate’ in implement the development highway**
Development Highway – Early 2005

Eclipse Platform

- Workbench
  - Source Code Resources
- Workspace
- Help
- Team
- Debug

Platform Runtime

- Eclipse Java Dev. Tools (JDT)
- IBM/Rational Dev. Tools (WSAD5)
- Service Design Tool (DynaRep)

Requ. Mgmt Tool (OCRWeb)

Select UML Modelling Tool (Component Architect)

Serena Version Control (Cmn ZDD)

Serena Version Contr. Tool (Cman DS)

Oracle DB

Fujitsu DB

ZDD Repository

DS Repository
Global Bank Developer Front End View (Q3 2005)

Eclipse Platform

- Workbench
  - JFace
  - SWT

- Workspace
  - EMF Ecore
  - XMI, RAS
  - Resources

- Help
- Team
- Debug

Platform Runtime

Eclipse = Strategic Client Integration Platform

- Eclipse Java Dev. Tools (JDT)
- IBM/Rational Dev. Tools (RAD)
- Innovations <tbd> Rules Modeler Tool (Visual Rules)
- Global Bank Service Design Tool (DynaRep)
- IBM/Rational UML Design Tool (RSM)
- Serena Req. Mgmt Tool (RTM)

Serena Version Contr. Tool (Cman DS/ZDD)
Global Bank Development Repository/Model View

Eclipse Platform

**User Workspace**
- SWC1
- SWC10
- SWCxx
- RW

**Team**

Load, Unload, Synchronize
Checkin, Checkout
Baseline

Global Software Asset Workspace

<table>
<thead>
<tr>
<th>SWC1</th>
<th>SWC2</th>
<th>SWC3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- SWC10
- ...
- ---

ChangeMan DS Repository

**EMF.CodeGen**

**DynaRep**

Oracle DB

**Data Sync**

**IQGen Code Generator**

**Model Project**
- DevelopmentProject1
- DevelopmentProject2
- ModelProject1

**Model Project**
- Eclipse Ecore
- Eclipse EMF
- Eclipse UML2
- Meta1
- Meta2

**Eclipse Platform**

**Global Bank Development Repository/Model View**

15
Summary: Global Bank use of Eclipse for Integration

- The complete Bank model data is available in a structured set of Ecore based model files grouped around Eclipse Projects. This builds the Master Repository of the Bank Model Data.

- A subset of the overall model data will be synchronized into an Oracle based Database. The information will be made available via "Webservices" to the DynaRep Eclipse Frontend for fast lookup and query requests.

- DynaRep FE has the overall knowledge which model item can be found in which Eclipse Project and in which Ecore File.

- Code Generation will be based on two lines:
  - IQGen JSP Template driven against Oracle based model
  - EMF.CodeGen

- In essence the bank is creating a Software Development Platform by aggregating and integration commercial and custom developed Eclipse tools to address gaps in the end-end lifecycle
Using Eclipse, EMF, UML…to build an
open, integrated tools platform

Many of the ideas – influenced by MDA and MDD

But pragmatically implemented at eclipse.org and its member companies
How is the IBM tools platform evolving

- IBM began the move to the use of EMF as a tools integration metadata platform in 2001
  - We had a couple of iterations as we started with early versions of the OMG MOF 1.3 spec
  - Several optimizations led to creation of EMF and the Ecore model
  - Over time the Ecore and MOF2.0 models have converged
    - The MOF2::EMOF compliance point is now supported by EMF (Import EMOF models to drive EMF generations)
- As we implement the Rational ‘Atlantic’ tools we are learning more about how to improve the Eclipse tools platform
  - It is becoming more model driven!
  - In addition to EMF support, we are beginning to add RAS support to desktop tools, then repositories
- The next few charts illustrate our current thinking
  - We expect this will iterate as new Eclipse projects (Web Tools Project, Data Tools Project, the anticipated ‘Graphical Modeling Framework’ GMF projects move forward
  - EclipseCon Technical Exchange had sessions by Borland, IBM and others to discuss the GMF project
IBM Modeling Platform Architecture: A simple view

- Product Level UI (actions, dialogs, wizards, editors,...)
- UML Diagram Plug-ins
- UML Notation Model
- Diagram Framework
- Notation Model
- Modeling Service Layer
- org.eclipse.uml2
- org.eclipse.gef
- org.eclipse.emf

Potential for diagram interchange
Metadata integration
IBM Modeling Platform Architecture: A simple view

Integration with Eclipse
Eclipse-independent core APIs
Compatibility with EMOF
Customizable serialization/deserialization to/from XMI
Interoperability with other EMF-based components

Model Interchange

Potential for diagram interchange
Metadata integration
IBM Modeling Platform Architecture

- Integration with Eclipse
- Eclipse-independent core APIs
- Common look and feel with other GEF-based components
- Extensible palette with built-in tools, ...
- Extensible controller framework for mapping model to view
- ...
- Potential for diagram interchange
- Metadata integration
IBM Modeling Platform Architecture: A simple view

Integration with Eclipse
- Eclipse-independent core APIs
- Compliance with UML (Unified Modeling Language) 2.0, an OMG standard
- Validation rules based on OCL constraints
- Interoperability with other UML2-based components

Potential for diagram interchange
- Model Interchange
- Metadata integration
IBM Modeling Platform Architecture:

- Service infrastructure
- Command infrastructure
- Icon Service, Parser Service, Properties Service
- Global Action Service

- Validation Service
- EMF Query and Indexing Service
- Batched notifications support
- Copy/paste support
- Semantic procedures

Potential for diagram interchange

Metadata integration
IBM Modeling Platform Architecture

Provides an extensible EMF-based notation meta-model
Optimized for GEF
Optimized for team support
Enables separation of semantic and notation models
Provides basic styles (font, location, color, etc...)

Potential for diagram interchange

Model Interchange

Metadata integration
IBM Modeling Platform Architecture

Realizes the notation model
Exposes a set of extensible services that allow you to create extensible diagrams
Reusable edit parts, edit policies and figures
Provides standard actions (e.g. Z-Order commands, fonts, etc...)
Provides clipboard support (SVG, etc...)
Provides linked diagram support
Reusable layouts
Print preview support
... Visualization service
IBM Modeling Platform Architecture

Extends generic Notation Model with UML specific styles e.g. Stereotype display mode
IBM Modeling Platform Architecture: A simple view

1 or more plugins per diagram type
- Actions
- Commands
- Decorators
- Dialogs
- Edit Parts, Edit Policies, Figures
- Icons
- Preference
- Properties

Potential for diagram interchange

Metadata integration
IBM Modeling Platform Architecture: A simple view

Model Interchange

Potential for diagram interchange

Metadata integration
Application Life Cycle Integration Platform
A call to action to the Eclipse Community

Language Tooling
(J2EE, Web Services, Deployment)

Data Tools
(RDBMS, XML...)

MDD Tools
(Object, Data Modeling, Code generators...)

Domain Specific Tools/Apps...

End to End Application Lifecycle Tooling (Eclipse.org member value add tools)

Eclipse Tools Integration platform (Models, APIs, XML formats...)

Web Services (XSD...)

J2EE (EJB, JSP...)

MDD/MDA (UML2,U2TP...)

Testing TPTP

EMF

GEF

JDT/CDT

RCP

ETC.

Eclipse Core

Code/Artifact Repositories, Management Tools (Eclipse.org member value add tools)
Summary

- Application Life Cycle Management is becoming a visible initiative driving software tool suites
- Eclipse and Visual Studio are anchoring the emerging tools platform and moving to support Rapid Application Development & Model Driven Development
- The Eclipse community is using open standards and open source to create a diverse and innovative tools ecosystem
  - Come join these efforts to simplify application development and deployment
- The Model Driven Tools Platform is ready to be exploited by vendors as well as enterprise customers
- EMF (MOF, XMI) and RAS are emerging as key influencing technologies for data integration and asset management for application lifecycle tools
Trademarks

- OMG™, MDA™, UML™, MOF™, and XMI™ are trademarks of the Object Management Group
- IBM™ is a trademark of the IBM Corporation
- Java™ and J2EE™ are trademark of SUN Microsystems
- XML™ is a trademark of W3C
- All other trademarks belong to their respective organizations
Thanks