



IBM Software Group

# Leveraging UML 2.0 and Open Source Technologies to Build Next Generation Modeling Tools and IDEs

Kenn Hussey  
March 23, 2005

Rational software



@business on demand software

# Contents

- What is Eclipse?
- Benefits of Using Eclipse
- What is EMF?
- Benefits of Using EMF
- What is GEF?
- Benefits of Using GEF
- What is UML2?
- Benefits of Using UML2
- Anatomy of a Next Generation Modeling Tool



# What is Eclipse?

- A kind of universal tool platform
- Open, extensible IDE for anything and nothing in particular
- The basis for a variety of tools and products



# Benefits of Using Eclipse

- Facilitates seamless integration of tools within and across different content types and tool providers
- Supports
  - ▶ both GUI and non-GUI-based application development environments
  - ▶ the construction of a variety of tools
  - ▶ an unrestricted set of tool providers
  - ▶ tools to manipulate arbitrary content types
- Runs on a wide range of operating systems



# What is EMF?

- *Eclipse Modeling Framework*
- A modeling framework and code generation facility for building tools and other applications based on a structured data model
- Provides
  - ▶ tools and runtime support to produce a set of Java classes for the model
  - ▶ a set of adapter classes that enable viewing and command-based editing of the model
  - ▶ a basic editor
- Provides the foundation for interoperability with other EMF-based tools and applications



## Benefits of Using EMF

- *Integration* with Eclipse
- Eclipse-independent core APIs
- *Compatibility* with EMOF (Essential MOF), an OMG standard
- *Customizable* serialization/deserialization to/from XMI
- Edit command framework for undo/redo support
- *Extensible* validation framework
- *Interoperability* with other EMF-based components



# What is GEF?

- *Graphical Editing Framework*
- Allows developers to create a rich graphical editor from an existing application model
- Provides
  - ▶ a layout and rendering toolkit for displaying graphics
  - ▶ many common operations which can be extended for a specific domain
- Employs MVC (model-view-controller) paradigm which enables simple changes to be applied to the model from the view



## Benefits of Using GEF

- *Integration* with Eclipse
- Eclipse-independent core APIs
- *Extensible* palette with built-in tools
- Handles for resizing objects and bending connections
- *Extensible* controller framework for mapping model to view
- Edit command framework for undo/redo support
- Common look and feel with other GEF-based components



# What is UML2?

- *Unified Modeling Language 2.0*
- An EMF-based implementation of the UML 2.0 metamodel for the Eclipse platform
- Provides
  - ▶ a useable implementation of the metamodel to support the development of modeling tools
  - ▶ a common XMI schema to facilitate interchange of semantic models
  - ▶ test cases as a means of validating the specification
  - ▶ validation rules as a means of defining and enforcing levels of compliance

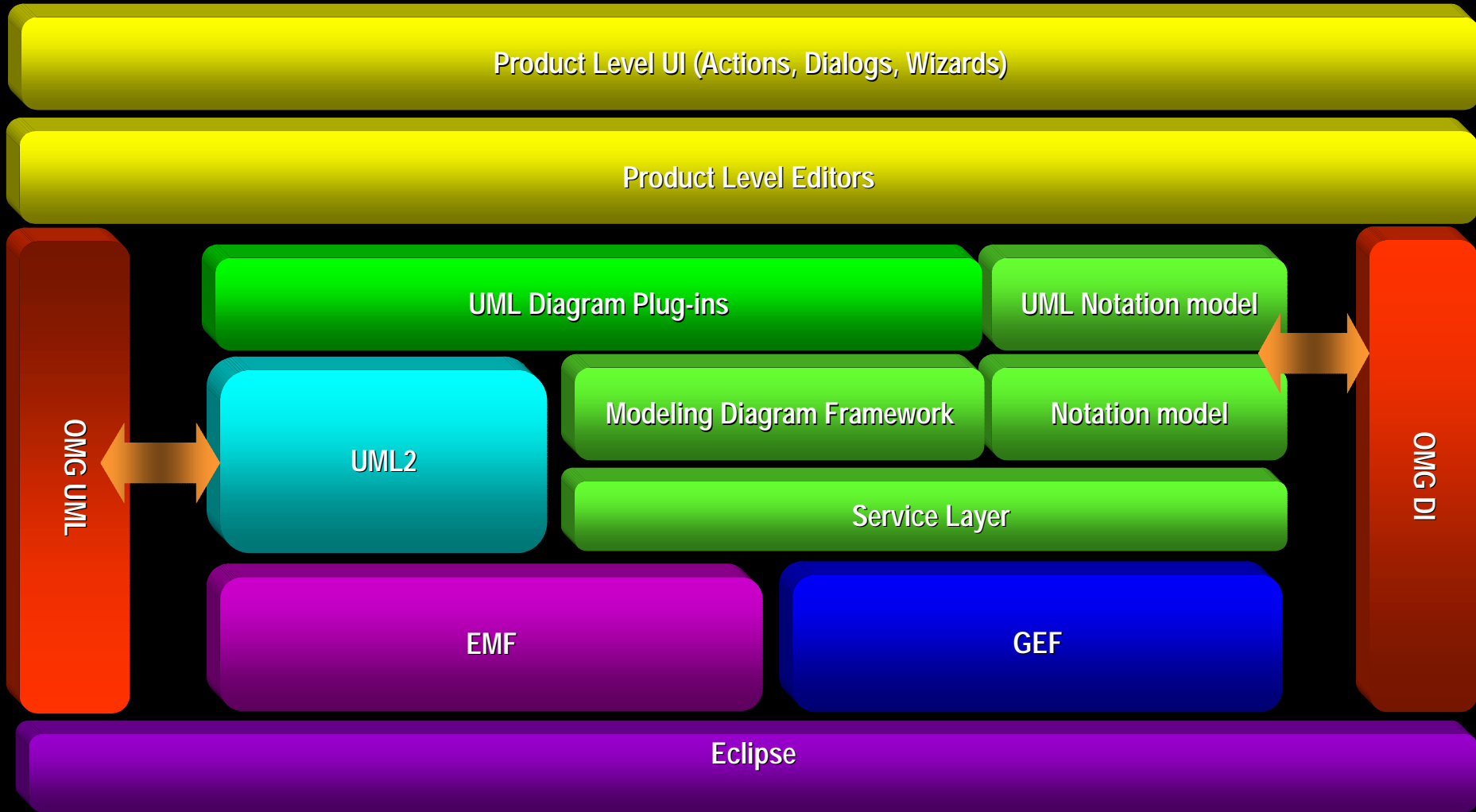


## Benefits of Using UML2

- *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



# Anatomy of a Next Generation Modeling Tool



THANK YOU





IBM Software Group

# Backup Slides

**Rational** software



@business on demand software

## UML2: Overview

- An Eclipse Tools subproject at <http://www.eclipse.org/uml2>
- An EMF-based implementation of the UML 2.0 metamodel for the Eclipse platform
- Project lead is Kenn Hussey ([khussey@ca.ibm.com](mailto:khussey@ca.ibm.com))
- Users include IBM (Rational Software Architect) and Omondo (EclipseUML2)



# UML2: Goals

- Provide a useable implementation of the metamodel to support the development of modeling tools
- Provide a common XMI schema to facilitate interchange of semantic models
- Provide test cases as a means of validating the specification and implementation
- Provide validation rules as a means of defining and enforcing levels of compliance



# UML2: Challenges

- Since the specification isn't finalized, the API is a moving target
- Aspects of the standard metamodel (redefinition, subsets/supersets) make the mapping to Java and XMI difficult
- There are a number of errors in the specification which, until fixed, make the API and XMI schema difficult to use
- Compliance points for UML 2.0 are a subject of much debate within the modeling community



## UML2: Status – Metamodel

- EMF-generated API based on a collapsed version of the metamodel in which package merges and a majority of the redefinitions have been factored out
- Includes mechanisms to support derived union, redefines, and subsets constraints
- Generated using a customized Ecore “builder” and JET templates



## UML2: Status – Interchange

- Default XMI schema generated using EMF
- Defines one namespace (<http://www.eclipse.org/uml2/1.0.0/UML>) for all elements, thus supporting interchange
- Elements identified using Universally Unique Identifiers (UUIDs)
- Will support import/export from/to OMG XMI for UML 2.0 once the specification is finalized



## UML2: Status – Specification

- UML 2.0 Superstructure FTF has delivered final draft and report to OMG
- Errors discovered while using the API and/or XMI schema are now being fixed and submitted as issues to the UML 2.0 Superstructure and Infrastructure Revision Task Force (RTF)
- UML 2.0 Superstructure and Infrastructure RTF already has a number of open issues, available at <http://www.omg.org/issues/uml2-rtf.open.html>
- Finalization date for UML 2.0 Specification is TBD (soon!)



## UML2: Status – Compliance

- Validation rules evaluated using EMF validator mechanism
- Validators implemented based on sets of OCL constraints
- Correspond to compliance levels



## UML2: Milestones

- Maintenance release 1.0.2 to coincide with Eclipse 3.0.2 and EMF 2.0.2
- Release 1.1 to coincide with Eclipse 3.1 and EMF 2.1
- Release 2.0 to coincide with Eclipse 3.2 and EMF 2.x



# UML2: Development Plans – 1.1

- Theme: Release Currency
  - ▶ Eclipse 3.1 / EMF 2.1 Compatibility
  
- Theme: Built To Last
  - ▶ Migration Framework
  - ▶ Resource Localization



# UML2: Development Plans – 1.1 (Continued)

- Theme: Simple To Use
  - ▶ Update Site Support
  - ▶ Instance Creation Support
  - ▶ More Examples
  - ▶ EMF Generator Extensions
  - ▶ Improved Documentation



# UML2: Development Plans – 2.0

- Theme: Release Currency
  - ▶ Eclipse 3.2 / EMF 2.x Compatibility
  
- Theme: Completeness
  - ▶ UML 2.0 Conformance
  - ▶ UML 2.0 Interchange
  - ▶ Validation Rules
  - ▶ Unit Tests
  
- Theme: Built To Last
  - ▶ Reduced Memory Footprint



# UML2: Development Plans – 2.0 (Continued)

- Theme: Simple To Use
  - ▶ Resource Fragments
  - ▶ Derived Features
  - ▶ Javadoc
  - ▶ Improved Convenience Methods
  - ▶ Enhanced Icons
  
- Theme: Broadening The Community
  - ▶ Tools



# UML2: How Can You Help?

- Develop tools based on UML2
- Report bugs
- Participate in newsgroup discussions
- Write articles
- Become a UML2 contributor!

